THE UNIVERSITY
of EDINBURGH

Capture and Generalisation of Close Interaction with Objects

Peter James Sandilands

Doctor of Philosophy
Institute of Perception, Action and Behaviour
School of Informatics
University of Edinburgh
2014

Graduation date: 29th June 2015

Abstract

Robust manipulation capture and retargeting has been a longstanding goal in both the fields of animation and robotics. In this thesis I describe a new approach to capture both the geometry and motion of interactions with objects, dealing with the problems of occlusion by the use of magnetic systems, and performing the reconstruction of the geometry by an RGB-D sensor alongside visual markers. This 'interaction capture' allows the scene to be described in terms of the spatial relationships between the character and the object using novel topological representations such as the Electric Parameters, which parametrise the outer space of an object using properties of the surface of the object. I describe the properties of these representations for motion generalisation and discuss how they can be applied to the problems of human-like motion generation and programming by demonstration. These generalised interactions are shown to be valid by demonstration of retargeting grasping and manipulation to robots with dissimilar kinematics and morphology using only local, gradient-based planning.

Acknowledgements

This thesis would not be complete without thanking all the people involved in teaching and supporting me during my time at the University of Edinburgh. I must first thank my supervisor, Taku Komura, for his support and advice provided during my time here.

Special thanks goes out to all my friends and collaborators that I've worked with and learnt from, both formally and informally, over the years. I'd especially like to mention: Joe Henry, Adam Barnett, Vladimir Ivan, Steven McDonagh, Bas Boom, Xi Zhao, He Wang, Kirill Sidorov, Myung Geol-Choi, Sergio Orts-Escolano, Dmitry Zarubin, Florian Pokorny and indeed all who came to Friday IPUBs.

The support of my family has also been important to me, I wouldn't be here without them, so a further thanks to my Mum, Dad, and sister Madeleine for their kind words.

Finally, my partner Alison Bennett has been instrumental in keeping me going and sticking at the work. She has been with me throughout my doctoral journey and I couldn't have wished for a better companion.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Some of the chapters that comprise this thesis are based on work performed during the creation of jointly-authored publications. These published works are:

- Peter Sandilands, Myung-Geol Choi, and Taku Komura. Capturing Close Interactions with Objects Using a Magnetic Motion Capture System and a RGBD Sensor. In *5th International Conference on Motion in Games*, November 2012.

- Peter Sandilands, Myung Geol Choi, and Taku Komura. Interaction Capture using Magnetic Sensors. In *Computer Animation and Virtual Worlds*, July 2013.

- Bas Boom, Sergio Orts-Escolano, Xi Ning, Steven McDonagh, Peter Sandilands, and Robert Fisher. Point Light Source Estimation based on Scenes Recorded by a RGB-D camera. In *British Machine Vision Conference*, September 2013.

- Peter Sandilands, Vladimir Ivan, Taku Komura, and Sethu Vijayakumar. Dexterous Reaching, Grasp Transfer and Planning Using Electrostatic Representations. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, October 2013.

- He Wang, Kirill A. Sidorov, Peter Sandilands, and Taku Komura. Harmonic Parameterization by Electrostatics. In *ACM Transactions on Graphics*, October 2013.

- Peter Sandilands and Taku Komura. Model topology change with correspondence using electrostatics. In *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology (VRST '14)*, November 2014.

(*Peter James Sandilands*)

Table of Contents

Chapter 1

Introduction

The main topic of this thesis is:

"How can we intuitively instruct robots and synthetic humanoids to perform dynamic interactions with arbitrary objects, such that these interactions can be reproduced on similar objects in novel situations and environments?"

which touches on considerations of a human-computer interface for interactions (i.e. interaction capture), techniques for processing these captured interactions to generalise them and then application of these generalised interactions for reproduction of the interaction (motion transfer). This is a broad topic, and so the thesis is limited in scope to the investigation of reach and grasp transfer on arbitrary objects and between varying manipulators. This is explored by preservation of the spatial relationship between manipulator and object.

1.1 Motivation

The manipulation of objects is a well-studied area in the fields of animation and robotics, as well as psychology and neuroscience. Despite this, there is not yet a global generative model of the manipulation of objects for arbitrary tasks and situations. Humans are able to generalise over multiple situations with few demonstrations. When manipulating an object, a change in its shape or location would not pose a difficulty to the person in carrying out the task. Previous experience provides an insight in how to tackle novel situations. However, generation of these grasps and manipulations by algorithm proves to be difficult, therefore this thesis focuses on addressing this challenge. Even by limiting the objects considered to the class of non-deformable objects, their properties may vary in shape, surface texture, colour, dimensions, utility, mass and moment of inertia,

to name only a few of the most salient. For a system to have human-like capabilities it must have the ability to classify the object whilst each of these properties vary; find correspondence between a known model and the novel one; and then adapt the motion accordingly. This means that generation of motions of interaction, such as grasping, or of manipulation of objects can be challenging when considering not only simple shapes but real-world functional classes of objects that can have significantly different properties. This challenge in robotics and animation is one that this thesis will explore: the ability to generalise interactions with objects to novel situations.

However, the generalisation of interaction with objects is difficult from a motion generation perspective, as the dimensionality of control is large for manipulators which have human-like degrees of freedom, and the constraints of collision and application of forces are complex and uncertain. Therefore this thesis looks at simplifying the task by capturing human motions of grasping and manipulation, and retargeting the interactions to both new objects and new manipulators, such as might be found in robotics, in a natural manner suitable for animatronics or teleoperation. I take a data-driven approach to this problem, first capturing data of interactions with an actor and objects, secondly finding an abstraction that captures the intention of the actor's hand well relative to the object, and thirdly adapting this motion to new objects or new manipulators by retargeting in this abstraction space and mapping to the configuration space of the manipulator.

I am interested in this problem as the task is particularly challenging, as it combines high degrees of freedom, natural human motion, and a large amount of contact between the hand and objects in a scene. It also requires adapting motion to novel situations. This generalisation of motion appears to be a promising direction for robotic interaction in new scenes, in that instead of planning motion from scratch, example interactions can be applied dependant on the task. There are two clear benefits to the reuse of previous successful examples: the simplified programming of a task (one can perform the task to demonstrate it to the robot), and also reduction of the search space when achieving a goal in a novel environment, as previous example motions are adapted to the new scene.

There are some difficulties with using existing techniques for this task. One such problem is that global path planning techniques for grasp transfer such as Rapidly-exploring Random Trees by LaValle [1998] and Probabilistic Roadmaps by Kavraki et al. [1996] are ubiquitous, and these have drawbacks. Although these are mature techniques they tend to generate 'robotic' looking motions, and can be slow when both

the degrees of freedom are high and the computational cost to validate a sample is large. Providing the correct constraints to give human-like motion in complex situations has proved difficult, and when adding these to already high degree of freedom tasks slows the solver down even more. Another aspect is the representation used during planning: often the representation is defined by what is simple for control of the system rather than what may be useful in the task. For grasping, this is often a representation which defines the configuration of the manipulator, and a more complex cost function is then sampled to find final grasp poses, which then requires path planning from the current pose by search in the configuration space. A more complex representation such as the spatial relationship between the hand and the object, or the envelopment of the object by the hand may lead to simpler planning requirements, as the planning then occurs in the relationship description, which has fewer boundaries and simpler paths to goals. I would postulate that humans do not reason purely in terms of joint angles when performing a manipulation of an object, rather they look at relationships and patterns in situations. This should be reflected in the approach taken when natural solutions are required.

Inspiration in finding solutions to generating motion of interactions could come directly from examples from actors. The capture of accurate close interaction with objects is important for observation of human solutions to tasks, but can be problematic. Many motion capture techniques work on the concept of optical data being captured by cameras, for instance reflected light from markers, or visual data which can be segmented and analysed. These have the intrinsic problem of requiring line-of-sight, and when close interaction occurs there are many occlusions. Due to this, often capture of close interactions with object requires a large amount of manual work to clean up the data, and also to recreate the object geometry.

Driven by these challenges and motivated by the transferral of natural human motion, this thesis looks to present a method for motion capture of close interactions, a representation that considers the spatial relationship of points and surfaces surrounding an object (and that encodes the concept of envelopment), and a system for dexterous reaching, grasp transfer and planning that is suitable for teleoperation and animatronics in novel environments.

1.2 Thesis Goals

This thesis looks to advance the field of motion transfer by considering novel representations of motion with relation to objects, by which the motion may be transferred to new situations whilst preserving its semantic style and 'naturalness'. It specifically looks at rigid and articulated-body object grasp and manipulation, and the transfer to not only new objects but to dissimilar manipulators. Throughout this thesis we define contact as a finite set of discrete point contacts using a linearised version of the Coulomb friction model for static contact. This model allows for static contact when the force that is normal to the object surface is less than $\mu \cdot N$, where μ is the coefficient of static friction between the finger and the object, and N is the normal component of the force enacted on the object by the contact caused by the hand. This can be considered as the normal force remaining within the friction cone generated from the contact between the two surfaces.

As control of more complex manipulators can have high degrees of freedom, this thesis investigates representations that allow for this motion such that paths in this representation can be planned without the use of global planners. Due to this, it needs to be a representation that ensures collision does not occur through linear interpolation, is valid in the entire configuration space of the manipulator, and represents aspects of the approach and grasp that are important in transfer. Furthermore, it must be robust to novel configurations: we do not wish to replan if the object moves. Therefore a novel spatial relationship representation based on the object surface must be defined which describes the motion of the manipulator relative to the object.

In order to achieve this, accurate motion capture with objects must be performed. As close interactions with objects can cause problems for current techniques, new techniques that preserve accurate relative motion between the character and object must be developed.

Finally the captured motion must be encoded in the relationship description and applied to new objects in the same class, whilst adapting the grasp on the object to ensure physical feasibility. This requires a technique for motion transfer and planning for manipulators based on the spatial relationship, which may transfer this relationship to new objects.

1.3 Contributions

This thesis makes the following main contributions to the capture and control of characters and robots for close interaction with objects:

- **Interaction Capture** A method and system for capturing close interactions with objects including object geometry, allowing for full reconstruction of the interaction for rigid bodies or articulated objects. A generated database of over 200 close interactions associated with this work is also contributed.

- **Electric Parameters** A novel representation of motion relative to an object that varies in a continuous matter (it both does not require discretization and provides a smooth mapping on object surface to outer space), forming an *object-centric coordinate system* as well as a geometric invariant for hand envelopment of an object.

- **Novel Applications of Electric Parameters in Graphics** A technique that produces an object with topology homeomorphic to a ball from an object of arbitrary topology whilst preserving the shape as closely as possible, and a technique that utilises the Electric Parameters to define a dense correspondence between objects that have topological holes or even disconnected components.

- **System for Reaching, Grasp Transfer and Planning** A method for motion transfer of human hands to robot hands of different morphology, via the representation defined in this thesis. Mapping motion to novel objects in a semantically and physically valid manner is also shown by utilising the object correspondence technique defined in this thesis. This gives a unified generalization of grasping and approach motion between objects of dissimilar geometry but of the same functional class, which allows for transfer of grasps to objects which may be grasped in the same manner for the same task. This system is limited to rigid, non-deformable objects.

1.4 Structure of the Thesis

Chapter 1 gives an overview of the problem, why it should be addressed and what the general contributions of the thesis are. Chapter 2 then follows with a review of the related work in the applicable fields. This thesis is then split into three main sections: Observation, Representation, and Motion Generation.

In Chapter 3 (Observation) the problem of capture of close interactions with objects is described, and a method to overcome this to obtain motion sequences that contain both the geometry of the model and the motion of the actor and model is presented. In Chapter 4 (Representation) an investigation of how the problem of motion abstraction and generation can be simplified by choosing an appropriate representation for the spatial relationship between the hand and the object is presented. Application of this representation to problems in the field of computer graphics is shown. Finally, in Chapter 5 (Motion Generation) it is shown how one can adapt the generalised manipulation motion to robotic manipulators and to novel objects using gradient-based planning techniques. These techniques allow for fast and robust grasp and manipulation transfer.

Chapter 6 then concludes the thesis and outlines promising directions for future work on this topic.

Chapter 2

Related Work

This chapter provides an analysis of existing work within the field of capturing motion of interaction with objects, relationship-based approaches to motion generation in animation and robotics, and grasp or manipulation planning and transfer. Work from the field of human anatomy is also considered.

This chapter has been structured to focus on three areas of the literature relevant to the thesis topic:

1. Motion Capture of Interaction - Focusing on the capture of human motion when the actor interacts with objects, and on capture techniques that specifically attempt to capture natural hand motion. The literature in this area relates most significantly to the work found in Chapter 3 (Observation).

2. Spatial Relationship Encoding using Alternate Representations - The various means by which alternative representations for motion and relationships between characters or objects in a scene have been encoded. Work in this field is most strongly applicable to Chapter 4 (Representation).

3. Grasping and Manipulation in Animation and Robotics - Previous techniques that have been utilised to generate motion of grasping and manipulation of objects. It is mainly related to Chapter 5 (Motion Generation).

2.1 Motion Capture of Interaction

The task of capturing the motion of close interaction with objects has recently attracted a number of research papers[for example Kry and Pai, 2006; Ye and Liu, 2012; Hamer et al., 2011; Mordatch and Todorov, 2012]. As close interactions often necessitate

manipulation or grasping of objects, animation of hands is necessary to progress this goal. Hand animation has been important since the early years of computer graphics [Catmull, 1972], and has increasingly attracted much more attention in recent years, as witnessed by the ever growing literature [ElKoura and Singh, 2003; Pollard and Zordan, 2005; Ho et al., 2010; Huang et al., 2011]. However, there is a lack of attention on capturing both the full object geometry and the hand motion whilst complex manipulations occur, which could assist in the understanding and reproduction of human-like manipulation. This thesis, therefore, seeks to develop novel techniques to capture these close interactions.

It is important to capture the full, dexterous motion of the hand as the interpretation of the scene can change even if subtle differences exist between the true motion and the captured motion [Jörg et al., 2010]. Because of the importance of this task, many methods for capturing examples of interaction have been created, with a variety of different goals such as accurate capture of the kinematic motion of the actor [Aristidou and Lasenby, 2010], capture of the partial object surface and fitting of a pre-generated model of the human hand [Hamer et al., 2011], capture of the contact forces between the hand and the object [Kry and Pai, 2006] or accurate capture of the human kinematics when close interactions occur without object data [Mitobe et al., 2006; Rahman et al., 2011]. However, despite the increased research activity in this field, and although multiple techniques have been proposed to capture scenes of human-object interaction, there is still room for improvement. Most have significant drawbacks in terms of estimation of the motion of the points in contact as they use marker-based systems that can become occluded (for example optical systems) or use mechanical systems such as the Cyberglove [Kessler et al., 1995] that attempt to capture joint orientations and not the locations of parts of the body, and do not capture free-body motions such as the motion of the object being manipulated.

My work combines the goals of many of these previous papers in the task of capturing the close interaction of the full body including the hand, ensuring no missing marker data whilst also capturing the full object geometry and motion in the scene [Sandilands et al., 2012, 2013a].

The following sections explore in greater detail the work relating to capturing hands or scenes of object manipulation in the following order: visual capture of the hands during interaction, marker- and mechanical-based capture of the hands, and, finally, magnetic motion capture systems.

2.1.1 Visual Capture of Hands and Interaction

A significant proportion of the work on capturing hand motion is focused on visual techniques, or a hybrid of vision and markers, in order to capture the fine detail of finger motion. These systems often analyse the visual input to find hand segments such that they can fit an internal model of the hand to the observation. A few techniques have recently attempted to capture the motion of objects in the scene, but require pre-known objects whose pose can easily be estimated using visual information (often using a pattern on the surface). The problem of identifying the hand's configuration is difficult and costly, so many techniques have attempted to alleviate this with control of the scene.

Control of the environment can simplify the problem of visual capture of the hand motion. Wang and Popovic [2009] describes a system that uses a coloured glove and a single RGB camera to reconstruct the pose of the hand for use in virtual reality applications. The only required hardware is a standard video camera alongside a glove with a custom pattern, making their system is easy to set up and low-cost. This patterned glove simplifies the pose estimation of individual sections of the hand, ensuring that interactive rates can be attained which they show with an experiment that allows the user to manipulate virtual blocks with their hand.

Wang and Popovic argues that marked-based motion capture is costly, intrusive and requires expertise to use, and that base-hand tracking (tracking without using markers or coloured gloves) is too computationally expensive to be used in real-time, so the only suitable method when needing a real-time, non-intrusive hand capture is to simplify the visual tracking problem by using coloured gloves or segments on the hand. Although this technique achieves adequate results for their intended purpose of being a low-cost and simple system for end-users to be able to easily capture hand motion, there are limitations to this approach: accuracy and the ability to capture interactions suffer from using cameras that can be occluded by objects. Wang and Popovic's solution only considers self-occlusion and so cannot be used when interacting with objects that may partially or fully cover the hand from the point-of-view of the single camera. This thesis, in contrast, prioritises accuracy and increasing the range of close interactions that can be captured accurately.

Nevertheless, the knowledge that can be gained from this contribution is that the coloured regions on the hand simplifies the task of pose inference to such an extent that the unique pose of the hand can be recovered from a single frame. This is done

by comparing the current image to a database of 18,000 generated (synthetic) images for known poses. As the glove has twenty patches of colour distributed evenly across the material, different poses appear significantly different (as opposed to attempting to determine the pose of the bare hand, where the variation in colour information is more subtle). When comparing the image to the database, the hand is segmented and resampled in a 40×40 pixel image. A Hausdorff-like distance[1] is used to compare each pixel in the query image to the database, and the identified pose is selected by blending the k-nearest neighbours given that distance metric. The system is shown to be accurate enough to be suitable for a user interface when performing coarse tasks, but results are not shown that provide the accuracy required when interacting with objects in a dexterous manner. Although interaction is shown in a virtual environment, capturing the geometry or motion of objects in the real world is not possible using this technique.

Using a glove is not always possible or desirable as it may affect naturalness in some motions. Cui and Sun [2004] describe another visual system that is used for hand tracking, this time tracking the bare hand, in which the objective is to use the motion in human-robot interactions. An experiment is shown with a user picking up a tennis ball with minor occlusions, which is then transferred into simulation. Images of the results look plausible in terms of accurate transfer, but no claims or experiments for evaluation are made regarding the success of the transfer, and only three fingers are shown in the example. The paper takes a 'Genetic Algorithm' approach to the problem, attempting to find a pose for the virtual 3D model that matches both the silhouette and edges that were detected from the input image by generating a large number of poses and combining these using selection, crossover and mutation operators. Experimental results qualitatively show superior performance over particle filter methods, which they explain by the failure of particle filter methods to scale well when the dimensionality of the problem is greater than 10 (pose estimation for the human hand being, in this case, a 26 dimensional problem). The accuracy of the method still appears to be lower than marker-based methods and as such, too low for the capture of close interactions. As with all purely visual techniques, occlusion requires estimation of the pose of the hand and so results in poor accuracy. Additionally, the object geometry is not captured nor is the motion of the object.

Learning from the more general field of hand animation also provides useful in-

[1] Informally: the furthest distance between an element in either of two sets to its closest element in the other set.

sights, with some research citing electro-mechanical or magnetic sensing devices as the most effective tools for capturing hand motion [Erol et al., 2007]. Despite this assertion, however, Erol et al. acknowledge that both systems have limitations: these systems are more expensive than visual systems, can affect the motion of the hand due to markers or gloves that must be worn, and often require calibration which requires additional time and expertise. Despite this, it is still the case that there are a number of challenges to using visual systems as the dominant form of capture. These challenges can be broadly encapsulated in the following three categories:

1. **Degrees of Freedom** - There are a large number of degrees of freedom in the human hand. Often the virtual model of the hand has between 24 and 30 DoF, not including the wrist translation, which means there are a large number of parameters to be estimated. Even with synergies between some of these DoF, for dexterous manipulation over a wide variety of tasks, at least six DoF are needed to perform complex manual operations [Santello et al., 1998]. To simplify this problem, researchers have looked at limiting the DOF of the hand when performing gestures (such as keeping the palm flat-on to the camera) with the justification that the user is attempting to communicate easily with the 'computer' [Quek, 1996]. However, in general this is too much of a constraint for natural interaction and manipulation of objects.

2. **Occlusions** - Occlusions between the camera and the hand can cause pose estimation errors as there is little information to go on when large sections of the hand are not in view. These occlusions can be self-occlusions due to the hand's pose, or object occlusions when interactions occur with items in the environment. This problem is a fundamental problem with vision techniques when interaction with objects is the goal: Although plausible motion can be generated by modelling the physical interaction with objects (such as shown by Wang et al. [2013b] and Oikonomidis et al. [2011b]), the pose is now estimated given some heuristics rather than captured.

3. **Complexity of Processing** - The algorithmic complexity is high with visual capture as often one or more images needs to be processed for each frame to extract features, and often a model needs to be fit to these features. Although algorithms on images can usually be parallelised, this is more complex and time-consuming for authors and requires careful design of the process.

Addressing the problem of occlusion, Erol et al. state that

> "If more flexible, non-posed, interaction is required (e.g., for object manipulation tasks), employing multiple cameras would be necessary." [Erol et al., 2007]

noting that object manipulation requires a more robust system that can deal with minor occlusions than the work up to that point had been capable of.

Wang et al. [2013b] mitigate the occlusion problem in this way. They address the need to improve motion-capture of close, dexterous interaction with objects by designing a novel method with multiple cameras. Several cameras are used to record a short sequence of motion of an actor manipulating an object, and then sampling-based optimization is used to fit a physically-based model of the hand and object to each of the temporally aligned videos. In this way, it is possible to reconstruct an approximation of the full physically-accurate motion of the hand and the object in the scene. Distinguishing this approach from previous work, the reconstructed motion includes the object, and so the input model of the object is supplied by the user as well as the physical properties (i.e. friction coefficient, mass). The main drawback to this approach, however, is the computation time: for a single frame the approximate kinematic tracking of the hand and object takes around 2.5 minutes, and to recover the parameters for the physical model a further 1.5 minutes are required per frame on current hardware. A secondary drawback is the restriction on the scene: many of the motions demonstrated were moderate in speed, and the environment was uncluttered. Further to this, because of the post-processing of the visually captured motion, there are differences between the interaction in the captured scene and the interaction that occurred. Despite this, this markerless capture method is impressive and appealing to users that wish to work unhindered by markers or gloves.

This two-step approach to motion reconstruction proceeds as follows. Firstly the kinematic pose is estimated per frame, considering the smoothness of the motion by penalising deviation from the configuration of the previous two frames (thus damping sudden changes in velocities). Secondly, given the target poses from the first stage, a proportional-derivative (PD) controller is used to calculate the internal joint torques to drive the hand's configuration to the target pose of the current frame. The object is not considered as actuated and so any motion of the object must come from contact with the hand, or from gravity which must be counteracted by the hand during the grasp. To ensure accurate contacts between the hand and the object (and so accurate motion of the object), a quadratic programming problem is created that tries to minimise the force at each measured contact point whilst ensuring the resulting force and torque

on the object is similar to the kinematically implied force and torque from the first stage. In some frames the contacts that are obtained via the PD controller following the kinematic targets are not suitable for the motion of the object (no matter how 'strong' the contact force is, the object may not move in the correct way). Because of this, the contacts are sampled around a set of poses that could possibly contain contact (i.e. if a finger is close to the object but not in contact, the state in which it is in contact must be considered as well as the state at which it is not). This accounts for some of the noise in the original kinematic pose estimation, as well as inaccuracy in the model as compared to the real-world, allowing valid contacts to be found so that the motion of the object closely matches the original motion, although of course increases the number of contact states per frame from 1 to 2^k, where k is the number of fingers that *may* be in contact. Wang et al.'s evaluation shows this second stage is essential for realistic motion as the vision system is unable to deal with occlusions and can pose the hand in a physically unrealistic manner if only collision detection is considered. Although the final motion appears physically plausible, the processing of the captured data takes significant time. It should be also noted that the object models are manually created and their physical properties have to be estimated correctly in order to ensure hand and object trajectories similar to the input motion.

Other techniques have used colour and depth (RGB-D) cameras for hand motion capture. Zhao et al. [2012] introduces a hybrid technique which combines optical marker-based motion capture and an RGB-D camera to improve the quality of hand motion capture compared to using only a single sensor type. This allows for the spatial accuracy of the visual system when self-occlusions are not severe, and the non-ambiguity and temporal accuracy of the optical system even when occlusion from the visual system occurs. The authors show that with 21 optical markers there are a large amount of occlusions in the optical system, and that on average only 15 are in view at any one time, noting that this is tedious manual work to label and fill in the gaps for the missing markers. This system uses both the visual and optical data by creating a cost function that represents the need to match the virtual marker positions on the hand model with the observed positions, ensuring that the virtual silhouette, depth and edges match with the RGB-D data. This cost function is combined with the hand kinematics and solved via particle swarm optimization (PSO). Results are shown using hand-only capture in a variety of poses from American Sign Language, grasping poses (although without the object present) and gestures. The results demonstrate that in certain situations combining the two sensor data ensures a superior result, although the algorithm

takes 1 second per frame and so is not suitable for real-time human-computer interaction. Furthermore, no examples are shown with objects and no object geometry or motion is captured.

Methods to use depth sensors together with vision sensors have also been utilised in capturing close interactions with devices, such as manipulating a mobile phone [Hamer et al., 2009, 2011]. This system uses structured light to obtain dense depth data, and fits each hand segment independently by considering samples in a local region around the previous state of the segment. The technique intentionally deals with the segments separately rather than as a single model to deal with occlusion in the scene. To ensure anatomical constraints are obeyed and the true hand structure is enforced, the probability of sampled independent segment transformations being true is multiplied by the probability of the segment transformation being consistent to the rest of the segments. There is manual intervention in the first frame when the complete pose is specified (96 DoF as each segment has a seperate 6 DoF transformation) and in some cases where the hand is occluded by the object, in which they manually label the position of finger tips in a key-frame fashion. As such, this technique can only be classified as partially-automated capture of interaction.

Another approach to deal with occlusions is to use the object shape to determine some constraints on the occluded hand. Oikonomidis et al. [2011a] investigated tracking the hand whilst interacting with objects by modelling occlusions and physical constraints, producing impressive results even when the hand is mostly occluded. The authors consider the manipulated object as a source of additional information regarding the hand pose, rather than a gap in truthful data about the hand, and so consider the object and hand in unison and seek the optimal parameters that best explains the observed *and* occluded data, penalizing intersection. They perform experiments on simple geometric primitives such as cuboids and cylinders in their work, but argue that arbitrary triangle meshes could be used "provided that this does not increase the dimensionality of the problem prohibitively"[Oikonomidis et al., 2011a]. They define the problem as a multidimensional optimisation problem in that they try to minimise the cost defined as $O(m,M) = \sum_{I \in M} D(I,m) + \lambda_k W(m)$ where m is the joint hand-object model, M are the multiple images recorded for a single frame from each of the cameras in the scene, $D(I,m)$ is the distance of m given a single image $I \in M$ and $W(m)$ is the cost of the penetration between the hand and the object. λ_k is a weighting factor. They define $D(I,m)$ as the discrepancy between the pixels of a rendered version of the model of the hand m and the pixels detected as the hand in the image I, similar to the previously

discussed technique of Wang et al. [2013b]. By penalising the intersection of the hand and the object, Oikonomidis et al. enhance the physical realism of the scene, whilst requiring the object to appear in the correct place in the scene as it must occlude the hand in a similar way to keep the cost of $D(I,m)$ low. Similarly to Zhao et al. [2012] they use PSO[2] to minimise the objective function. Results are shown that appear to be fairly accurate and natural, but are only shown on simple objects: no item is more complicated than a cuboid. It should be noted that there are noticeable discrepancies between the visual image in real world sequences and the model, making the method suitable for HCI but perhaps not as useful for motion capture where the true, dexterous motion must be captured.

Section Summary Overall, existing research using visual information for hand capture has shown to result in high-quality data when the hand is not occluded. However, when required to manipulate and interact with objects in a close, dexterous manner (which results in significant occlusions) the capture results are less successful. This thesis seeks to address this issue. Some motions of manipulation and grasping are not feasible due to the inherent limitations in using a system which requires line of sight, such as reaching into a bag to pull out an object, putting objects into containers, dressing motions or even more complicated grasps that result in fingers or the object being occluded from any point outside the capture area. Although more 'open' interactions have been successfully captured with good results by Wang et al. [2013b], the processing time for even short motions is excessive: assuming 30 frames per second a single second of video takes 120 minutes to process, and a minute of recording takes 120 hours. For capturing any longer sequence of motion a user would likely be frustrated by this wait. Marker-based techniques look to provide solutions which are less computationally complex whilst retaining high quality motion. These techniques are reviewed in the following section.

2.1.2 Marker and Glove-based Capture for Hands and Close Interaction

This section looks at techniques which are mainly focused on markers, either optical, inertial or otherwise, for interaction capture. Mechanical systems (such as gloves) are also used for capture, but often need to be paired with a marker-based system to obtain

[2]Particle swarm optimisation

the wrist transformation, and so are also discussed here.

A common problem with marker-based motions is that the size of the markers affords only one marker on each of the finger tips. This can lead to problems when estimating the orientation of each section of the hand. Aristidou and Lasenby [2010] discusses motion capture with constrained inverse kinematics for real-time hand tracking and describe how they reconstruct full 3-dimensional hand poses using low-dimensional optical motion capture and inverse kinematics (IK) in real-time. They capture the hand using 6 markers: 5 on the finger tips and 1 placed on the back of the wrist. Because the markers are placed only on the finger tips and wrist, the joint configuration for the whole hand, including the intermediate segments of the hand whose position aren't directly captured, needs to be estimated. They do this using the FABRIK (Forward And Backward Reaching Inverse Kinematics) method [Aristidou and Lasenby, 2011] to iteratively solve the IK problem from this low-dimensional representation given by the points from the optical motion-capture. FABRIK first starts at the outermost joints (the end effectors, in this case the finger tips), and works up the kinematic chain to the root (in this case the wrist) adjusting each joint by attempting to find the joint locations that ensure the bone length constraints are respected whilst the end effector and root positions are closest to their targets. It then sets the root to the identified marker position and traverses the kinematic chain in the opposite direction, towards the outermost joints, adjusting the locations of the joints by solving the same problem. It iteratively does this until the distance between the end effectors and the target location is within some tolerance. Experimental results in the paper show that the method produces visually natural, biomechanincally correct motion and also prevents oscillations even at low frame rates ($\sim$ 10Hz). As their method uses only point constraints and a single marker at the wrist, the orientation of the entire hand is ambiguous in certain configurations of the palm. This limitation is not addressed in the work. Another, perhaps greater problem, is the assumption that all markers are visible at all times. This is not the case for optical markers when interacting with objects, and so this technique as a whole is not suitable for interaction capture (although the IK technique they use, FABRIK, could be adapted to other systems).

For movies and films the accurate capture of the hand geometry can be useful. Instead of capturing the skeleton and using a traditional method for skinning, Huang et al. [2011] reconstruct the geometry of the hand and its complex deformation under different configurations using dense laser scans and low-dimensional 'control points'. Experimentally the control points are either manually defined in a keyframe sense or

obtained via an optical motion-capture system using 18 markers for a single hand. The fine wrinkles and creases that occur due to hand postures are difficult to model either by an artist or by heuristic, so approaching the solution as a data-driven one is appropriate. The hand is automatically split up into more than a thousand patches on the surface which the authors call 'abstract bones', and the transformation of each of these bones is learnt per-pose given the input of the control points. Although the produced geometry at each pose appears natural, the authors show that the pose of the hand will not necessarily follow the marker positions exactly. These errors are a significant problem for interaction with objects, as accurate motion from the hand is required to obtain the correct relationship and contact points with the object. Although this was not the primary focus of Huang et al.'s work, it does mean this technique is not suitable for interaction capture, even ignoring the other problems of optical capture.

Kry and Pai were aware of the importance of interaction with objects and proposed a method for capturing the contact force between a hand and an object [Kry and Pai, 2006]. Their system captures the joint positions using optical motion capture, and the joint compliance[3] via linear force sensors and finger tip pressure sensors. Using this, they are able to physically simulate the captured motions on objects with different physical properties, such as friction coefficient and mass. This technique has the drawback of having to wear devices on the finger tips and so naturally affects the motion the actor produces. However it gives direct data regarding the physical forces in the interaction. A beneficial result of this is that the exact moment of contact can be easily determined from the force sensors. Humans take around 100ms to change muscle force due to reflexive actions (e.g. when they experience an unexpected perturbation) and longer when the change is intentional [Johansson, 1996], so by capturing these motions at 500Hz they are able to observe at least 50 frames of the compliance of the joint when it is perturbed, before the muscles activate to resist or adapt to the force. The compliance can then be found by dividing the derivative of the joint angle difference (obtained by optical motion capture) by the derivative of the torque (obtained by contact force measurement). As they use optical motion capture the technique has similar occlusion-based drawbacks as the visual systems when the interaction becomes more complex. The objects must have embedded force sensors as well, which may not always be possible. The approach presented in this thesis does not capture the forces of interaction, but instead focuses on capturing the geometry and motion of the interaction. From this, contact points can be estimated and overall force on the object can

[3]The inverse of joint stiffness.

be computed. Unlike Kry and Pai [2006], the technique does not limit the interaction to finger tip contacts, nor does it require wearing of devices on the underside of the hand where interaction takes place.

As capturing of high quality motions is time-consuming and requires experience, researchers have previously attempted to create open databases of everyday events using marker-based techniques. The CMU motion database[4] includes motions such as cooking, locomotion and playing basketball. These are close interactions with the environment, but do not contain the required detail for full reconstruction, such as finger movement as those markers are omitted, nor do they include the geometry or motion of objects in the scene. The supplied data is only of the raw markers rather than the whole body motion and so is not aligned to any object correctly. Also the interaction with 'obstacles' is restricted to full body obstacles such as uneven terrain and large items blocking a walking trajectory. Although for some takes video and audio data is provided, much manual work is required to create an accurate representation of the scene. Whilst this is a superb resource, it is not directly useful for detailed dexterous manipulation of objects.

Alternatives to Optical Systems Many of the optical methods have occlusion problems, but as Zordan and Hodgins [2002] discusses, there are alternatives to optical motion capture such as magnetic or inertial systems. To constrain the pose of the hand further given the captured information, markers which capture orientation as well as position can be used, giving a 6 DoF constraint rather than 3 DoF. Zordan and Hodgins [2002] investigate motion capture that can be applied to physically simulated characters using such markers. Their technique utilises wired magnetic motion capture for the upper body of the character, but does not capture the dexterous motion of the hand, choosing instead to capture only the wrist. This allows them to create a motion controller to simulate playing table tennis and other sports assuming a fixed relationship between the wrist and any object in the hand. The paper does not focus on the motion capture technique, however the motion is shown to be smooth and continuous: no occlusions occur as the magnetic markers do not rely on line-of-sight. The interaction with the object must be explicitly modelled by their system and is not captured (for example, the table tennis bat appears to be parented to the wrist at a fixed offset, similar to another bone in the kinematic model of the character), nor is the geometry of the object captured. Although not suitable for dexterous interaction capture, it shows the

[4] http://mocap.cs.cmu.edu/

promise of magnetic systems.

The Xsens system [Roetenberg et al., 2013] is an example of a commercial motion-capture system that uses both inertial and magnetic sensors to compute joint angles for the body. The system is limited to capturing humanoids as the process relies on computing the orientation of joints in the biomechanical model, and is not able to counter the drift from inertial sensors reliably enough to use to track objects' transforms independently. In tracking a character the estimation of the location of the root joint is aided by intelligent constraints added to the biomechanical human model in the Xsens software. For example, there is an assumption that if there is foot contact with the floor, the feet will be static in the vertical axis and there will be no drift of the root joint [Roetenberg et al., 2013]. Because of the biomechanical model they use and the combination of sensors of different types, the system can be used in magnetically noisy environments for short periods (up to 40 seconds) without loss of accuracy. The magnetic component of the system relies on the earth's magnetic field. Although this system contains a magnetic component it is unable to be used to generate independent position and orientation for free-floating objects, making it unsuitable for capturing of character-object interaction. The magnetic component of the XSens system is used to correct for the drift of the inertial sensors but can be disabled, leaving the inertial sensors alone to estimate joint orientation and position without correction. Inertial systems directly measure the linear acceleration and angular velocity using accelerometers and gyroscopes respectively. Obtaining the position and orientation of a marker can be done by double integration of the accelerometer data and integration of the gyroscope data over time, although this is only accurate for a few seconds due to numerical inaccuracy and integration drift [O'Donovan et al., 2007; Giansanti et al., 2005].

Practically it is often difficult to capture high-quality motion in everyday environments. Vlasic et al. [2007] discuss this and presents a system that uses ultrasonic and inertial sensors worn by an actor. They are unable to reconstruct the global transformation of the actor, but the resulting joint configurations are visually similar to the true motion. They also show that drift from the inertial system can reliably be reduced by combination with the acoustic sensors. Limitations to the system are that high-impulse motions such as kicking and jumping cannot be recorded, and interaction between multiple characters. Vlasic et al. also do not show finger motion in their captures. Because of these reasons alongside the lack of root position, their system appears unsuitable for reliable interaction capture.

Mechanical systems have been used to capture motions that have been difficult to

capture traditionally due to occlusion. Mechanical systems such as the 5DT Dataglove [Fifth Dimension Technologies, 2014] and the Cyberglove [Kessler et al., 1995] do not have any problems from occlusion, but instead suffer from the problem of encoding approximations of local rotations of joints rather than positions of end effectors. This leads to an accumulation of error along the kinematic chain, as well as a reliance on accurate calibration in order to obtain motion that is similar to the captured motion. This error, along with the common problem that not all joints on the hand will be captured[5], means that although the capture may preserve the style of the motion and be useful for human-computer interaction, the location of the finger tips are likely to be incorrect. This causes severe problems when dealing with interaction with objects as the finger tips are often used.

A mechanical system called the SPIDAR-8 Walairacht et al. [2004] avoids the problem of end effector accuracy by measuring the location of the finger tips using wires attached to each finger. This allows for the hand pose to be reconstructed using Inverse Kinematics, so a user can interact with objects in a virtual environment. It also allows for force-feedback as the wires attached to the fingers are actuated by motors. Manipulation of a virtual Rubik's cube is shown to be possible by this system. However, capturing interaction with objects in the physical world is not possible, as the wires prevent most motion in the open space around the hands, and may not be disturbed by external forces in order to ensure the accuracy of the reconstructed hand pose is maintained. This limits the use of this system and prevents capturing accurate interactions with objects.

In general, magnetic motion capture systems have the advantage over visual and optical techniques in capturing close interactions as they do not suffer from the occlusion problem, and they are superior to mechanical systems as they are accurate at the marker locations without occluding the workspace with many wires. Molet et al. [1999b] showed that a wide range of full-body motion can be captured using these systems, and that the ability to capture orientation with magnetic markers can be used to reconstruct a character's pose accurately. However, magnetic motion capture systems suffer from their own problems. Previous alternating current (AC) magnetic field systems [Burdea, 1993; Krieg, 1993] suffered from eddy currents which are induced by the surrounding metals. These currents produce their own smaller magnetic fields and thus the sensors receive a distorted magnetic field and the accuracy degrades. A

[5]The Cyberglove is one of the most popular mechanical glove captures systems, and captures only 15 joint readings, whereas the hand has over 22 degrees of freedom [Hashimoto et al., 2011] as discussed in Section 2.3.1.

common approach to magnetic sensor construction nowadays is to use a DC field. In the DC field approach [Blood, 1989; Ascension, 1998], a gap is produced between the timing that the transmitter produces the field and the sensors detect the field such that the eddy currents disappear. Fields are produced in three directions sequentially to obtain the 3D location of the sensors as well as their orientation in the environment. However, these DC sensors are less accurate and are negatively affected by the earth's magnetic field, power outlets and electric motors, which modern AC systems are resilient to. DC sensors also tend to be slightly larger and less accurate, meaning that mounting them on the finger tips is not comfortable for the actor and affects the naturalness of the motion. Furthermore, it has been shown that AC systems are suitable for capturing dexterous finger motions [Mitobe et al., 2006]. Mitobe et al. [2006] present a system that utilises a magnetic motion capture system for capturing a pianist's hand motion. The hardware[6] is physically modified for the wires to be more flexible so as to capture the subtle motions associated with playing the piano, attaching these sensors to the fingers. Although the hand motion can be captured using this technique, the object geometry and the interaction between the hand and the object cannot be captured. Nevertheless, magnetic capture systems appear promising, and one of the aims of this thesis is to produce a framework that does not suffer from occlusions. For these reasons this thesis utilises a magnetic (AC) system: the LIBERTY system by Polhemus Inc [Krieg, 1993].

Section Summary Each of the optical techniques share a drawback with visual techniques: occlusion. If a marker is occluded, the location must be estimated, and an accurate estimation relies on an accurate model of the motion. As there is no accurate generative model for dexterous motion, capturing these motions without occlusion issues for acquisition of high-quality motion continues to be an area demanding further research. The inertial systems described do not have the occlusion issue, but instead suffer from the problem of drift, meaning that the positional accuracy reduces as time goes by due to integration error of the inertial readings. It is notable that even a well-established system, the Xsens, does not give an interface to the estimate of position of free-floating objects in the scene (given by the passive magnetic sensors combined with the inertial sensors), such is the problem of drift. The occlusion problem can instead be mitigated by using an active magnetic motion capture system, which can still be considered a marker system as they record the location and often also orientation

[6]The Polhemus LIBERTY system, which is used in this thesis.

of markers attached to the body. Previous work has shown that high-frequency dexterous motion can be accurately captured by these techniques in certain situations. In Chapter 3 it will be demonstrated that by combining the magnetic sensors with captured geometry data of the object it is possible to perform true interaction capture that allows an analysis and understanding of the spatial relationships in the scene.

2.2 Spatial Relationship Encoding Using Alternate Representations

This section investigates recent methods that seek to describe or preserve the spatial relationship of a scene in order to cope with motion generation, mainly with re-targeting to new characters. Rather than focusing purely on the joint angles of characters or position of the joints, these methods look at some relationship between the character and the scene to store the motion. This section provides an overview of the techniques that have been used to try to represent motion as a series of spatial relationships and how the spatial relationships change as the animation progresses. This is especially pertinent to the work undertaken in this thesis as the intention is to encode the motion in an object-centric coordinate system and also describe the envelopment of the object by the hand.

There are many examples of work using fixed positional constraints when generating motions with contacts [Michelman and Allen, 1994; Kry and Pai, 2006; Liu, 2009; Mordatch and Todorov, 2012]. This contact representation is common as it allows for static grasping and physical interactions with the environment, with a simple point-based representation. For example, Yamane et al. [2004] explores an approach to animate characters manipulating objects using constraint based inverse kinematics, where the constraints are the position of the feet, position of the object, and position of the hands relative to the object. Some of these constraints (the positions of the feet on the ground and the position of the hands on the box) can be seen as spatial relationships between the character and the environment, with the constraint that the distance between these points must be zero. The authors use a motion capture database with similar constraints to bias the resulting poses towards previously observed poses, and explore the constrained pose space by Rapidly-exploring Random Trees (RRT) [LaValle, 1998]. RRT-style exploration is known to give unnatural transitions in motion and so Yamane et al. perform a postprocessing step of smoothing the resulting trajectories. However,

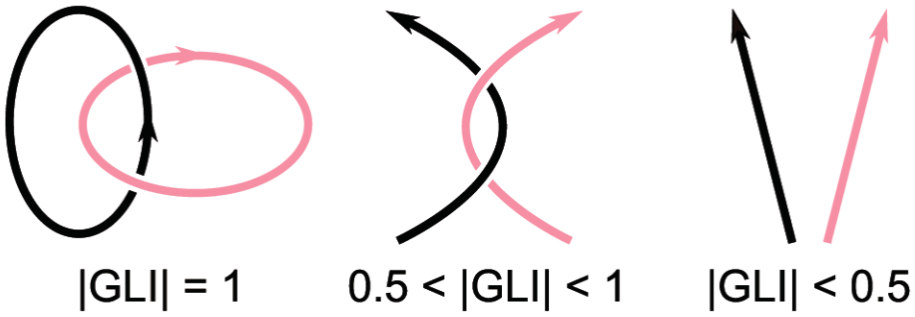


Figure 2.1: Examples of Gauss Linking Integrals between two curves, image from [Ho and Komura, 2009].

even after this step the motion does not appear fully natural [Yamane et al., 2004]. Their technique is not a fully automatic system as a user has to specify the constraints manually, however it does reduce the time taken to generate these motions from an hour for keyframed motion to less than 10 minutes in this system. Yamane et al. also notes that methods based on RRT do not scale well as the dimensionality increases as they are based on random sampling in the space. For motions of dexterous manipulation the degrees of freedom of the task are often too high.

Only slightly more complex than fixed positional constraints, simple distance and orientation relationships between the centroids of objects have been regularly used with some success in the generation, classification and retrieval of scenes [Ogawara et al., 2003; Merrell et al., 2011; Xie et al., 2013]. However, these relationships do not consider the surface of the object, and so can be problematic when close interactions occur.

Concepts of winding have also been used in representing relationships. Ho and Komura [2009] discusses the difficulty of planning a sequence of motion when the character must move between different intertwined configurations. For instance, if a character's arms are crossed in the first keyframe and a second keyframe is given such that the hands are then behind the back, a naive interpolation of joint states or positions of the joints causes self penetration. In these cases using RRT or other global planners can achieve a successful but unnatural sequence. However this results in a lot of random sampling of poses and 'wasted' computation as most samples will be useless. The authors instead propose a topology based approach that maps the generalised coordinates of the character to the tangle space represented by Gauss Linking Integrals (Fig. 2.1). This is termed by Ho and Komura as 'Topology Coordinates'.

By linearly interpolating in the topology coordinates, a successful knotting trajectory can be computed without collision, and this trajectory can then be mapped back to the configuration space of the character. Ho et al. [2010] applies this to two simulated Nao robots [Aldebaran Robotics, 2014] in order to transfer between different tangling poses. These ideas have been later used in the robotics literature as 'writhe' in order to encode the amount a kinematic chain wraps around a sequence of line segments [Zarubin et al., 2012]. Although these techniques are able to preserve the ideas of wrapping between curves, they do not investigate direct relationships between *surfaces* as will be explored in this thesis.

Instead of looking at curves a point-based approach is presented in 2010 by Ho et al. [2010], who defined the 'Interaction Mesh': a volumetric representation of the relationship between the joint positions of a character, two characters, or a character and an object skeleton. They use the Laplacian coordinates to define the location of the character's joints relative to other sample points in the space. When a single point is moved, the rest of the points in space must move to preserve the displacement energy[7] of the scene. To ensure a sparse set of relationships between nodes (joints and markers) in the scene, the Delaunay tetrahedralisation is used and only connected components have constraints. This means that only nodes that are nearby affect the relationship of each other directly, whereas further away nodes have a less predictable effect as they must act via their neighbours. This technique is later used in the robotics literature to retarget motion capture of dancing to robots whilst maintaining balance [Nakaoka and Komura, 2012], and to automatically adjust the configuration of a robotic arm at runtime using relationships to obstacle markers in the space [Ivan et al., 2013]. Similar ideas regarding relative vectors have been used for retargeting animation based on the surrounding environment. Relationship Descriptors [Al-Asqhar et al., 2013] were defined using the distance field around objects in order to retarget motion for character animation. They represent the character as a set of joint locations with bone length constraints and attempt to minimise the difference between the relative location and orientation of the joints given a set of sample locations on the surface of the environment when the environment is deformed. These are only suitable for motion retargeting and not for planning novel motions as sample points are fixed given the initial playback of the data.

Spatial relationships given character pose have also been investigated by Wang and Komura [2011]. They showed how local control can be used to solve interpolation

[7] The difference in the Laplacian coordinates summed over all the joints.

problems between sparse keyframes when the character must achieve complex poses. The overall concept is that of repulsive energy: a character whose body segments are near to each other has a high repulsive energy whereas a character whose body segments are far has a low repulsive energy. The authors noted that it is easier to interpolate linearly between poses when the body segments are far apart than it is when the segments are close to each other, and so proposed that in difficult keyframes the character can be controlled to minimise the repulsive energy in both keyframes until a linear interpolation is feasible.

Following on from this work, Wang and Komura [2012] use topological representations to control a cloth in the task of wrapping and knotting. As cloth has a large number of degrees of freedom in its particle representation, global path planning methods are not suitable for control. The cloth is instead controlled by moving a 'control line' on the surface of the cloth to a 'target line' on the object surface, passively moving the rest of the cloth particles using a cloth simulation. This reduces the dimensionality of the problem as the task becomes moving the 'next' particle in the control line to the correct position on the target line until it is within some threshold, repeating this until all particles are moved to their correct places. This technique lowers the active degrees of freedom by simplifying the task, only having to actively control a single particle at a time.

Zarubin et al. [2013] looked at a grasp metric defined by the approximate surface of a hand and the object by computing the integral of discrete Gaussian curvature over geodesic balls centred around vertices on a target object. This approximately encoded the idea of a perfect manipulator of a certain size being able to cage[8] an object, and searched for areas in which this caging value was large in order to grasp those locations. This metric was not valid for the outer space of the object, is not a continuously varying measure, and requires exhaustive search over the vertices of the object, unlike the coverage measure which will be defined in this thesis (discussed in Chapter 4).

Further consideration is given to surfaces covering a mesh by Igarashi and Suzuki [2011]. Their system creates customized dust covers for three-dimensional closed meshes by computing multiple convex hulls of a small number of subsets of the object vertices, and then takes the union to create a cover which envelops the entire object. They measure the approximate coverage of the dust cover during a test for the ability

[8]Wrap around an object such that the object cannot move arbitrarily far from the manipulator without penetration.

to remove the cover from the object, given an opening in the cover (the 'uncovering test'). This is performed using the object's convex hull, and only gives a binary 'covered' or 'uncovered' result, as it uses the difference of the normal vectors of the dust cover moving along the convex hull, and denotes the object uncovered when the differences are smaller than 90 degrees. Unlike the envelopment measure defined in this thesis, Igarashi and Suzuki [2011]'s technique is not continuous and is conservative in its estimation of coverage.

Section Summary All of the techniques previously employed in the field either do not handle relationships with general, concave objects, or do so by setting specific samples on the object surface from which then distance-based relationships can be drawn, meaning they rely on a discrete, sampled representation of the relationship and must select locations to place samples. Accurate dexterous grasping and manipulation often require exact contact locations and so require samples specifically at contact points: the required contact points may not be known when transferring these grasps to novel objects. Further to that, these motion generation techniques are all for retargeting based on examples, and do not deal with generation of novel motion when the examples are too different. Retargeting a grasp from an object to a new object that differs significantly may require replanning based on physical feasibility which none of these techniques consider. Because of this, existing techniques are not suitable to represent general grasping motions. Gauss Linking Integral and skeleton-to-skeleton relationships are interesting for representing continuous topological relationships; this thesis looks at extending these line-line relationships to surfaces (Chapter 4).

2.3 Grasping and Manipulation in Animation and Robotics

This thesis looks at transfer and generation of grasp and manipulation motion. It is therefore important to consider the literature on grasp analysis and generation in robotics and animation, hand animation, and programming by demonstration.

2.3.1 Grasp Analysis

Since the early 1900s, researchers have been interested in understanding human grasping and manipulation of objects [MacKenzie and Iberall, 1994]. Initially this was from a medical perspective, as after World War One interest in prosthetic hands was high

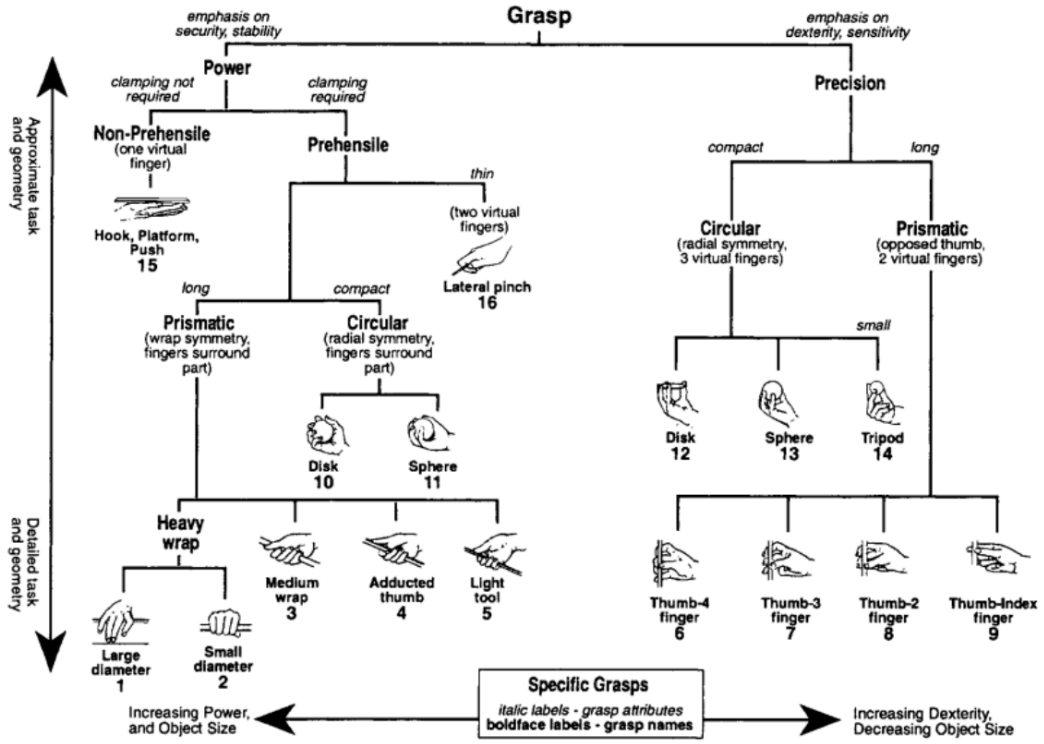


Figure 2.2: Cutkosky and Howe [1990]'s hierarchical decomposition of classes of grasp.

due to the number of injuries sustained [Slocum and Pratt, 1946; Taylor and Schwarz, 1955].

In early work on grasp classification, a small number of main grasps were identified, later being expanded to a wide variety dependent on task: spherical, lateral, finger tip, hook, palmar and cylindrical are all common examples, and more can be seen in Fig. 2.2. Many of these reoccur through the literature although there has been much disagreement about how fundamental each is. Napier [1956] proposed a popular description of grasp styles still used today, segmenting the space of grasps into *power* and *precision* dominated by properties of the task and hand configuration. He showed that movements of the hand consist of two basic patterns of grasps:

- *Power Grasps* are grasps which make maximum contact with the object, in order to keep the object stable, and allow the user to exert their full strength for manipulation of the object. The object remains clamped between the flexed fingers and the palm, with opposing force being provided by the thumb.

- *Precision Grasps* are grasps usually made by the finger tips for finer control over the object's motion, often between the thumb, index, and middle finger. Having

fewer contacts allows for in-hand manipulation.

Prior to Napier other medical researchers defined three fundamental units of function in the hand: grasp (which in this paper meant a power-grasp style grip), pinch, and hook [Slocum and Pratt, 1946]. Further investigators produced subdivisions in the main grasp styles, creating terms such as prismatic, circular, and tripod grasps [Elliott and Connolly, 1984; Lederman and Klatzky, 1987; Cutkosky and Howe, 1990]. Later on, the concept of "virtual fingers" were introduced, in that a group of fingers can be combined into a single virtual finger when they work together in a task [Arbib et al., 1985]. This abstracts the actual configuration of the hand from the manipulation and instead looks at the functional components of the forces applied on the object to the grasp, reducing the degrees of freedom in the analysis. These were then used to investigate grasp quality; notably the authors of [Iberall et al., 1986] define the term "opposition space" to mean "the area within the coordinates of the hand where opposing forces can be exerted between virtual finger surfaces in effecting a stable grasp.", describing the intuitive understanding that opposing forces on an object assists stability of a grasp. Each of these grasps and concepts have been shown to be important for description of grasp style, and a full grasp generalisation and transfer should be able to achieve all of these grasps in order to simulate the natural motion of the human hand as closely as possible.

Work has also been performed on reducing the dimensionality of grasping for encoding the natural synergies in human motion (by finding the latent variables) but also but simplifying the representation of the human hand pose. In 1998, Santello et al. produced a paper investigating synergies between the degrees of freedom of the hand when using tools [Santello et al., 1998]. The technique employed here used the Cyberglove to measure the joint angles of the fingers directly (which has the assumption that the Cyberglove does this both accurately and fully, which may not be the case) and use discriminant analysis to maximise the ratio of the inter-group variance to the intra-group variance of a linear combination of the joint angles when grasping 57 different objects. This paper famously argued that $> 80\%$ of the variance in pose comes from the first two principal components of motion of the hand, but many that discuss this paper omit that this variance is in the readings of the Cyberglove which has only 15 DoF (whereas the hand has at least 22 degrees of freedom [Hashimoto et al., 2011] and can be modelled to have up to 27 [ElKoura and Singh, 2003], or more when considering the deformation of the palm [Yasumuro and Chihara, 1997]). However, it can be clearly seen that there are indeed synergies between the individual muscle groups in

the hand, and in fact there are dependencies on certain joint configurations given other joints [Weiss and Flanders, 2004; Latash et al., 2007]. This is partly due to mechanical coupling and interference by the muscles and tendons in the upper arm [Taylor and Schwarz, 1955]. It is likely that for natural motion of the hand, a means by which these synergies can be emulated is vital.

Kang and Ikeuchi [1992] proposed an idea for a simplified representation of grasps and contact relationships between the hand and an object called the *contact web*. Here the hand is represented as a grasp connecting the five fingers to the palm via the distal, medial and proximal phalanges[9].Contact is defined between single points on a finger segment and the object, augmented by the object normal and 3D location of the contact. This is used in later work to attempt to recognise grasp types automatically by using the concept of virtual fingers along with a grasp hierarchy to determine the grasp based on relative locations of fingers on the object, the relationship of the fingers to each other, and their normals[Kang and Ikeuchi, 1994].

Inspiration taken from methods for the study of grasp quality from the field of robotics have also been used in the study and reproduction of human hand motion in animation. For example, the authors of [Ciocarlie et al., 2009] investigate finger tip contact point locations on objects during grasping and manipulation using the 'Grasp Wrench Space'[10]. They discover that only some people change grasp locations (and so affect the Grasp Wrench Space) when they know a disturbance will occur, showing that behaviour during object manipulation is dependent on the style of the character. Low dimensional representations of manipulation have also been proposed. [Steffen et al., 2008] captured human motion whilst removing a bottle cap and discovered latent parameters by using a unsupervised kernel regression on the captured 26 DoF hand pose to create a manifold of valid postures.

Section Summary The hand is capable of a wide range of motion with high degrees of freedom. There are complex relationships between the motion of the joints, with joint limits being dependent on the current pose. That being said, a limited set of motion can approximate the hand pose for a variety of tasks. Dimensionality reduction in this way has proved to be useful to simplify the problem of motion generation, as will be discussed in the following sections regarding animation and robotics. Many papers seek to classify the relationship between segments of the hand and objects during

[9] Finger segments.

[10] This will be described in Robotic Grasp Planning (Section 2.3.3).

manipulation, with a focus on finger tips as much of the fine manipulation of objects occurs there.

2.3.2 Generation of Grasping and Manipulation Animation

This section looks at the most relevant work in the fields of computer animation and graphics for generation of grasping and manipulation of objects. In these fields, the drive is often for the appearance of natural motion, so as that the viewer believes the world the characters exist in, but also for artistic control over this motion. These considerations manifest themselves often in requirements for parameters to tune the motion, and (in interactive situations) the ability to adapt to novel situations. Natural motion can be captured from an actor in the original environment: we make the assumption that a *natural* motion for a task is a solution to the task that would not be uncommon for a biological actor to take. If a motion is captured of a biological actor performing some task, it is natural as they are performing the task. These motions are only guaranteed to be natural in the original environment however, by placing the digitised version of this motion into new scenarios it is required to adapt the motion to different objects, terrain and other characters in the scene.

Papers in grasping and manipulation in the field of computer animation can broadly be categorised into the following main groups:

- **Geometry-based, Model-driven** Systems of this type are based on the geometry of the object and an internal model of the hand grasp style, with kinematic control (e.g. [Sanso and Thalmann, 1994]).

- **Geometry-based, Data-driven** These systems also rely on the kinematics of the hand and shape of the object, computing grasps via properties of the geometry of the model rather than the forces at contact locations. They are data-driven, directing the search for these grasps based on examples (e.g. [Aydin, 1999]).

- **Physics-based, Model-driven** These systems use physical simulation to determine the grasp, either by computing the overall forces required by an example or by measuring how well the hand is able to grasp the object. They have a heuristic model which they use to generate the grasps (e.g. [Liu, 2009]).

- **Physics-based, Data-driven** These systems are similarly physics-based, using forces and torques to control the hand for interaction with objects in the scene, but instead of using a generative model only they use motion capture or example

data in order to direct the search for a good grasp (e.g. [Pollard and Zordan, 2005]).

Recently, hybrid approaches which combine an initial geometry or model-based motion with physics-based processing are becoming more popular in an attempt to mitigate the drawbacks of the above approaches whilst bringing the strengths of both, usually at the cost of additional implementation complexity. For example, [Wang et al., 2013b] uses data in capturing a motion, a kinematic phase to fit the hand and object trajectory approximately, then a physics-based refinement stage in which they sample contacts between the fingers and the object. As was noted by Wei et al.,

> "Physical validity is not a sufficient condition of natural motion"[Wei et al., 2011b]

and these approaches seek to allow for both kinematic capture and dynamic refinement given physical laws. This section considers these hybrid techniques to be a natural progression of physics-based approaches, and so groups them together.

Geometry-based, Model-driven Approaches Earlier work focused on final grasp poses and using IK to reach target locations. [Bekey et al., 1993] explains the development of a kinematic grasp planner, considering robot hands with multiple fingers. [Sanso and Thalmann, 1994] created a model-based system for grasping sphere, cylinder, and block primitives based on simple rules and IK. [Huang et al., 1995] describes a heuristic method that decides approach and grasping strategy, for example pinch or lateral grasp, based on the object geometry and 'multi-sensors' attached to the hand for collision detection[11]. Similarly, Rezzonico et al. [1995] focuses on wrapping the fingers around an object using a finite-state machine (FSM) approach, ensuring that a grasp directed by a gloved actor does not penetrate objects in a virtual scene. They do this by fast collision detection with the virtual object, adapting the directed motion to the object's geometry. The free hand, grasping, and grasped states are detected, allowing the object motion to be directed by the hand once grasped. Boulic et al. [1996] proposes a technique that allows for a grasp on a virtual object to be modified such that the fingers may move and manipulate the object by the fingers during the grasp (known in robotics as *in-hand manipulation*). It seeks to overcome the limitation that the object is rigidly parented to the wrist post-grasp, which occurs in many model-based techniques. It does this by taking into account the finger motion and considering

[11] A sensor is an object attached to the hand, usually spheres, often used for fast collision detection and as triggers to change grasp state.

the Coloumb friction model for contact with the object. The motion of the object can then be generated by the user in the hand, although the motion isn't necessarily physically accurate. Each of these rule-based systems are comparatively easy to implement and fast, but tend to work only for a limited set of objects such as geometric primitives, may not work in situations with physical interaction as physically-based grasp metric aren't used, and are unlikely to produce natural movement in a wide variety of situations.

Koga et al. [1994] take a motion planning approach via discrete optimisation to generate animations of manipulation of a single object between two configurations. They only require that the initial pose and constraints on the object are specified in order to plan the multi-arm motion that moves the object kinematically to the target location. The planner works over a discrete selection of target locations in the effective work space of the character, with the joint angles of the arms discretised in 0.05 radian intervals. They do not consider the dynamics of the motion, nor do they consider the continuous space of motion, resulting in the planned motion being neither dexterous nor physically plausible. However, the ability to set general tasks of manipulation by merely specifying a few constraints using this technique is useful.

Other techniques that instead focus on specific tasks have been presented in the field of hand animation, with the idea that creating heuristics or models for a small class of interaction is simpler than a general model, and can produce higher-quality results. For example, the task of playing a guitar has been considered[ElKoura and Singh, 2003]. Guitar fretboard fingering is a complex task that takes humans years to master. ElKoura and Singh [2003] present a method for automatically animating a human hand to select the correct configuration for playing the guitar given input 'tablature' (fretboard locations for each finger). The postures can be unusual but still physically valid, which presents challenges when attempting to realistically replicate this in animation. The authors note that assumptions such as $\theta_{DIP} = \frac{2}{3}\theta_{PIP}$ (the distal interphalangeal joint rotation is two-thirds of the proximal interphalangeal joint rotation) is inadequate for these extreme postures. They mention joint re-targeting, though it is limited to using the joint angles and so does not take into account any spatial relationships. The conclusions of ElKoura and Singh supports the assertion in this thesis that creating complex hand motion is time-consuming and difficult to do manually, and that even domain-specific solutions such as the 'Handrix' system presented by them have problems with naturalness because the subtle motion the hand exhibits when performing complex actions is hard to encode in a heuristic algorithm, and may

be much easier to capture from data.

Geometry-based, Data-driven Approaches Gleicher and Litwinowicz [1998] uses data to alleviate the problem of unnatural motion due to unrealistic heuristics. Their paper describes a method for retargeting motions from motion capture to new characters, including picking up an object and moving it. The captured motion is not dexterous, and so deals only with wrist locations. The technique works only for characters with identical skeleton hierarchy but is able to preserve contact location on the object when the character's size changes. [Aydin, 1999] also addresses the issue of animating grasping of arbitrary objects using a data-driven approach. However, they segment the reachable space into subvolumes, and capture or precompute primitive grasps for each of these. Upon being directed to grasp a novel object, it is first classified as one of the primitives, then the hand pose is given by the nearest subvolume. A version of IK is then used to perform the grasp. None of the approaches described so far can be considered dexterous, as they either perform a simple autoclosure computation in which the fingers close around the object, or they only work on a small number of primitives using unnatural heuristics.

Further planning approaches have been used to generate manipulation motions between characters in a continuous manner, combining data-driven motion with constraint-based IK [Yamane et al., 2004]. Using a library of captured motions they are able to direct the search to realistic sequences whilst respecting the non-collision constraints with the environment and posture constraints set by the user. To fill in minor gaps in the motion library, IK and a model of human motion is used. Only simple grasps were demonstrated in the examples, and they cannot be considered dexterous.

A database of grasps has been used to generate novel grasps on objects [Li et al., 2007]. Li's paper takes the final grasp poses given to the system and produces similar poses ranked by quality on new objects. It considers a pre-made 'Hand Pose Database' that gives information about the hand joint configuration and also the contact points on some original object that was grasped. These contact points can be matched to new object geometries, giving a ranking of different hand poses for a certain object. They finally randomly sample triangles (made up of three contact points from the hand pose) and align each of these to the vertices on the object surface. A grasp quality metric can be computed from this information, determining how well the inner surface of a hand fits new geometry. However, Li et al. does not consider the approach to the object and leaves the natural grasping motion problem to a separate motion synthesis stage that

they do not address. Interestingly they consider some of the relationship between the hand and the object in terms of the surface normals at points on contact, allowing for some generalisation of the grasp by matching some attributes of the contact surface to new objects. This aspect lends support to the idea that the relationship between the hand and the object is important in grasping.

The authors of [Hamer et al., 2010a],[Hamer et al., 2010b], and [Hamer et al., 2011], describe a system to estimate an interaction with an object using visual systems (as discussed in Section 2.1.1), and then to generate new interactions on objects by preserving the relationship in the coordinate system of the object. This means that objects have to be closely aligned and similar to the original object model, but if they are they can warp the pose of a grasp to novel objects. The technique proposed in this thesis has similar goals, but approaches the problem by considering the spatial relationship with the surface of the object, rather than by only considering the object's centroid and rotation. This allows the object-centric coordinate system put forward in this thesis to adapt to the object deformation and does not require the object to be as similarly oriented and shaped. Hamer et al.'s technique also focuses on the tracking of motion, whereas this thesis takes a different approach to motion capture and can retarget to manipulators with novel kinematics.

Physically-based, Model-driven Approaches Automatic synthesis of object manipulations such as hammering a nail and pouring a glass of wine are demonstrated in [Liu, 2008]. The authors use physical simulation and simple kinematic goals to generate hand-object interactions that respond to disturbances effected by the user. The user supplies a grip style, object description, and task description in terms of what object should end up in which location or functions of the hand position and its derivatives. For the approach to the object, they use the approximation that often the hand imitates the shape of the surface of the object, and so they calculate the final grasp pose for a larger, closer version of the object and shrink these contacts until the surface of the object is grasped. This is an interesting approximation, but as they note it does not necessarily work for a wide range of objects, such as objects with handles for hook grasps, or objects with triggers and similar protrusions that may require a finger to grasp behind them.

The authors propose further methods of synthesis of close interaction based on optimization and contact constraints. Liu synthesizes motions such as opening the top of a bottle by specifying the initial grasp pose and contact points between the finger tips

and the bottle top [Liu, 2009]. The range of movements that can be synthesized by such a method is limited to those mainly involving the finger tips. This work takes an optimisation-based approach to generating motions of manipulation of objects. Starting from a given grasp pose, their method automatically computes the torques in the fingers to perform a task which is given by the motion of the object being grasped. The system can also locally change the fingers in contact with the object to withstand perturbations and provide a greater range of possible forces on the object. Their method doesn't require any predefined movement of the hand: this is both a benefit in terms of simplicity, but also a drawback as the resulting motions can be unnatural. As mentioned in Section 2.3.1, it can be difficult to programmatically encode the strategies that humans use in manipulating objects due to their high DoF and the muscleoskeletal system that humans posses. Although Liu's system is physically based it has to make some concessions in order to make the problem tractable, and these concessions can cause unnatural motion such as resisting a perturbation by solely increasing the joint torque without changing the pose. Unlike the system proposed by Liu, the data-driven methodology proposed in this thesis is able to generate human-like approaches and grasps due to the use of motion capture for the wrist and fingers.

The range of movements that can be synthesized are extended to those involving the palm in [Ye and Liu, 2012], in which the wrist position and orientation, alongside the object transformation are captured from an actor and then the finger movements are synthesized. This technique relies on optical capture and so does not work in situations where occlusions of the wrist or body may occur. Whilst impressive results are obtained, comparisons with captured finger motion in open manipulation examples show discrepancies and penetrations if other objects are in the scene, and the authors note that the fingers sometimes move unnaturally. Furthermore, the computation time is non-negligible as expected from spacetime optimisation, with examples shown taking over 10 minutes to produce 120 captured frames.

Other systems have looked at generating motion which allow contacts with any part of the virtual body. Contact-Invariant Optimisation [Mordatch and Todorov, 2012] has been used to generate grasping and manipulation motion from high-level goals on the object specified by the user. They generate a cost function based on the idea that contacts can occur at distance, but with a penalty. They perform the optimisation in two stages that update the trajectories 'key states'[12] with increasing cost for the contact distance and physical violations at the second stage. The objects they deal with are all

[12]Samples at every 0.5 seconds.

capsules, meaning that projection onto the surface using the distance field does not lead to problems as it is harmonic in this case (i.e. it has no local extrema). Concave meshes may lead to the optimisation failing, however these cases are not discussed. This work is interesting due to the limited input required in defining tasks, but as of yet has not been shown to produce true physically-valid or natural motion.

Natural hand models based on physical realism or biomechanical models have been proposed for improvement of the visual quality of generated hand motion. A realistic hand model lends plausibility to an animation, making investigation of natural hand models relevant to the subject. Albrecht et al. [2003] did not consider grasping, but instead the complexity of the human hand and generating a realistic model of both the motion and the appearance, based on modelling the anatomical detail. Animation of the model is controlled by virtual muscles, which affect the deformation of the skin using a mass-spring system. Visually the appearance is natural, but interaction with objects isn't possible using this technique, and accurate collision detection with the non-convex mesh would be slow and possibly unstable using current techniques. Because of this, many models use approximate convex or primitive geometry for collision detection, and finer geometry for rendering.

Yasumuro and Chihara [1997] consider a virtual spring model of hand actuation in which the skeleton of the hand has varying stiffness at each joint dependent on the virtual springs attached to them. These virtual springs bring the hand back to a neutral pose when some torque is applied to the joints, with the resisting torque termed the inner energy. Manipulation and grasping isn't directly considered. The target neutral pose is set manually, which may be difficult to do due to the expert biomechanical knowledge required of the relationship between different joint rotations.

Biologically inspired models of the human hand have also been proposed in order to improve naturalness in pose and reduce the motion space that is plausible given the degrees of freedom in most joint-based models[van Nierop et al., 2007]. Work in animation has also gone into improving the hand model for natural motion and rendering. Musculotendon simulation for hand animation [Sueda et al., 2008] looks to simulate the biological structure of the hand in order to get natural bulging and shrinking of the skin under certain poses. These techniques help in the visual appearance of the hand but do not deal with object manipulation.

Physically-based, Data-driven Approaches Pollard and Zordan [2005] presented an early example of both a data-driven and physics-based system for hand animation.

Physics-based systems can appear natural as they are physically correct (within the assumptions that are used in terms of hand geometry and abilities), and so the interactions in the world appear plausible. The effort of the animator then is redirected from preserving the plausibility of the scene into creating the controller of the hand, which is usually complex and unintuitive to define, and makes it difficult to predict how the hand will act in new situations. This paper demonstrates that, whilst creating the controller for a task is possible and that physical parameters can be estimated by simple techniques such as the 'drop test'[13], results appear unnatural in the case of grasping more complex objects and for more complex tasks.

Another technique that deals with both capture and motion generation, and was described previously in the related work (Visual Capture of Hands and Interaction, Section 2.1), is presented in the paper by Wang et al. [2013b] that explores capture of hand interaction with objects, as well as retargeting to novel objects. The results shown are of high quality and physically realistic, however there are limitations to the technique: the hand must be mostly visible to the six cameras, and the motion is post-processed from what is observed to make it approximately physically accurate. This post-processing removes some of the qualities of the original motion in favour of motions that are easy for the algorithm to generate, losing some of the benefits of motion capture over novel motion generation. It is also important that the environment is fairly easy to segment out during capture in this technique, limiting the environments that capture may take place in. For adaptation of the motion to new objects, a Particle Swarm Optimisation approach is taken to sampling contact points which provide the required forces to the object. As mentioned earlier, this approach is slow: a single second of motion at 30 frames per second takes around 120 minutes to generate.

Zhao et al. [2013] uses a pre-recorded motion database and an RGB-D sensor to allow a user to generate grasp and manipulation of objects by acting them out in open space. It uses physical-based simulation to generate these motions from a large, manually-labelled dataset of captured grasps in realtime. They show examples of grasps on convex objects, which suit this method as their grasp quality measures restrict the grasps to ones that centre around the centroid of the object. The objects are able to be manipulated after the grasp completes by the motion of the wrist. A limitation in this work is that precomputed grasps are stored relative to an object's local coordinate system, which means they do not directly adapt to the surface geometry

[13] Where a user manually tunes stiffness and related parameters whilst an object is dropped on an outstretched hand: in making it appear natural, realistic physical values are set for the hand.

when performing the approach and grasp but rely on scaling the objects and collision detection. Despite this, the results are impressive.

Section Summary Data-driven motion retargeting tends to produce higher-quality motions than model-based systems, as they capture subtle movement details that are difficult to define in a constrained model or controller. Successful physically-based manipulation of objects has been achieved via motion planning but tends to take a significant amount of time to generate the motion and the results are often unnatural, especially in the approach to the object. Recent approaches have increasingly looked at realistic retargeting of captured motion, but none have tried to describe the envelopment or coverage of a grasp in an abstract way, nor have they looked at more complex spatial relationships with the scene than point-based relationships. Furthermore, transfer to characters with different morphology or kinematics is rarely discussed. Robustness to novel objects is often a problem too, as many previous techniques require the novel objects to be similar to the original object and manually aligned. Many approaches use space-time optimisation and sampling, which is slow for these problems with high degrees of freedom. The limitations in existing research discussed here are addressed in this thesis, through development of a techniques that allows motion transfer to novel objects without manual mapping or alignment, and to novel manipulators that differ morphologically.

2.3.3 Grasping and Manipulation Planning in Robotics

For a robot to interact with its environment, the robot needs to be equipped with a suitable internal representation, enabling it to plan the actions required. The problem of reaching and grasping, in particular, is a crucial component of many interactions. Differently to animation, the motion generated does not necessarily have appear ‘correct’ to a viewer, but must be able to robustly produce the required change in the environment required by the task. Robots for entertainment may also have the constraints of Section 2.3.2, making this a challenging task. There is utility in generating motion that a viewer would expect to non-entertainment robots as well: humans that are required to interact with these robots would be able to intuitively understand the tasks the robots are attempting, making collaboration easier. It is also reasonable to assume that robots attempting to achieve a goal when interacting with objects used by humans can use human-like motions to create a suitable solution for the task, which may be taken as a

starting point for refinement into an optimal solution.

Several approaches and representations have been proposed for the purpose of generating manipulation and grasping motions. These are reviewed below.

2.3.3.1 Grasp Planning

In robotics, the field of Grasp Planning looks at automatically selecting poses and joint torques that grasp objects. Of particular interest is *dexterous* grasping, which in robotics traditionally means something slightly different to its use in everyday language: the ability of a hand to change the position and orientation of a manipulated object arbitrarily in the workspace[14]. To understand the field, the quality measures that have been proposed for ranking of grasps are important to consider. Two important concepts surrounding grasp quality in robotic manipulation are *force closure* and *caging*.

Force Closure The notion of force closure [Murray et al., 2006] is a classical physics based modelling approach to the grasping problem, where a grasp is formalized as a collection of point contacts between an object and the robot's manipulator. A grasp is *force closed* if wrenches[15] in arbitrary direction can be counteracted by the robot. To evaluate the force closure condition, the robot needs to be able to estimate contact points and normal directions at each contact point.

The L^1 grasp quality Q metric proposed by Ferrari and Canny [1992] is a popular approach to evaluating the force closure criterion under the assumption that forces can be independently controlled at each contact point. Miller and Allen [1999] extends the original 2-dimensional idea of force closure to 3-dimensional space by considering the contact location and normals between the manipulator and the object with the assumption that a force at the contact location can be produced as long as it remains in its friction cone. The Minkowski sum[16] of the wrenches produced by concatenating the force and torque vectors at the contacts is the 6-dimensional construct known as the Grasp Wrench Space (GWS). This conceptually describes all the possible wrenches on the object that the grasp can resist (remembering the assumption that the manipulator is able to actuate all forces in the friction cone of the contacts). A metric can then be produced for the magnitude of the minimum wrench the grasp cannot resist, by con-

[14] The area in which a robot can act.

[15] A wrench is the concatenation of force and torque at a point.

[16] $\oplus : A \oplus B = \left\{ \vec{a} + \vec{b} | \vec{a} \in A, \vec{b} \in B \right\}$

sidering the hyperplane closest to the origin. This is dubbed the ε quality measure. A large variety of such grasp scoring functions have been developed in recent years [Suárez et al., 2006], some of which are purely heuristic, while others take a more physics-based approach. To determine a stable grasp, random sampling and a ranking by such a quality measure are typically used in state of the art simulation environments such as GraspIT [Miller and Allen, 2004], OpenGRASP [León et al., 2010] and SIMOX [Vahrenkamp et al., 2012] which then allows the highest scoring grasps to be executed on a real robot with confidence. A change in object geometry then requires the robot to reinitialize its grasp synthesis. In my work, the focus is instead on transferring stable grasps under variations of configuration of the scene and parameters of the character and object.

Alternatively, the search for an optimal grasp has been formulated as an optimization problem [El-Khoury et al., 2012]. Since scoring functions such as Q typically depend only on local information about the object at the points of contact, their optima are highly dependent on a good reconstruction of the object. Since force closure in particular is a concept which depends only on the *local geometry* of an object around the contact points, it is sensitive to local noise. This dependence has recently been made precise in terms of the Lipschitz continuity of Q[17] [Pokorny and Kragic, 2013] which allows one to bound how much Q changes if error bounds on contact point, normal, and centre of mass estimates are known. Attempts to alleviate this have looked at computing minimum grasp quality over regions of the object. Reachable Independent Contact Regions (rICR) [Roa et al., 2011] is a technique for measuring the quality and robustness of a grasp in a real-time fashion, suitable for assisting a user in choosing grasps on an object by clearly displaying the robustness to perturbations and noise in the form of multiple contact points on the object that allow for a grasp of at least some minimum quality. This technique addresses the concern with point-based grasp quality metrics that the grasp quality can change significantly when the contact points change slightly, and in real-world robotics it is unreasonable to expect the planned contact points to be the exact contact points between a robot and an object in the real world. It does this by using Independent Contact Regions (ICR) [Roa and Suarez, 2009] which takes in an initial grasp of quality Q and a threshold value α where $0 < \alpha \leq 1$ that defines the minimal quality of the grasp relative to an initial grasp, and returns a series of labelled points on the object such that any contact can move between points in its own region whilst maintaining a grasp quality of at least αQ. rICR extends the technique to only

[17] Which states that a function is limited in how fast it can change.

consider grasps which are possible for the manipulator in question, and so computes the valid workspaces for the finger tips offline, then uses these workspaces to limit the search space for ICR online. Approaches related to force closure constitute a mature approach to robotic manipulation which has been well developed over the last three decades [Bicchi and Kumar, 2000].

Caging The *caging* of an object, on the other hand, is an alternative approach depending on the global geometry of an object and hand configuration. Rather than determine grasp stability by contact points and friction, it considers the object as a rigid body and determines whether a grasp is successful by ensuring that, as defined by [Kuperberg, 1990], an object X in the plane is caged by a finite set of points in the plane if it cannot be moved arbitrarily far away from those points without collisions. While this definition can be extended to 3D and to caging by arbitrary shapes in place of points, most work on caging to date has focussed on the 2D case and on analytically provable cages. The notion of stretching and squeezing cages in 2D, in particular, was studied by Rodriguez et al. [2011] and caging configurations are considered as a step towards a fixating grasp.

Considering caging in 3D is a relatively recent addition to the existing work in this field, and early work in this area has informed the development of this thesis. The work of Diankov et al. [2008] proposes a planning approach for using caging grasps for tasks such as opening doors and drawers and argues that caging grasps allow for a larger degree of robustness since a caging grasp does not enforce brittle rigid constraints and allows for a larger range of motions compared to a force closed grasp. For objects with holes, Pokorny et al. [2013] have proposed an approach which synthesizes caging grasps by controlling a hand via a gradient based approach involving winding numbers and by planning grasps using a basis of shortest homology generators. This work was extended by Stork et al. [2013] using a notion of virtual linking quantifying the amount of interlinking between loops on an object with holes and the robot hand's fingers and includes a task-space RRT to plan the motion of a robot arm.

In the work of Zarubin et al. [2013], a scoring function for the synthesis of likely caging grasp points based on geodesic balls is introduced. Since no analytic solution to the caging problem is currently known for general configurations of objects and manipulators in 3D, this grasp heuristic can be used to select appropriate grasp locations when force-closure grasps either aren't appropriate or necessary. For example, universal grippers[Brown et al., 2010] do not have fingers and so classical grasp planning

using force closure is not suitable. Instead, caging can be used successfully.

For practical grasping that simply considers the stability of the object grasp, pre-grasp poses can be used based on the shape of the object. Huebner and Kragic [2008] uses box-based decomposition of objects and applies a box pre-grasp shape to the hand, attempting to approach the object from the unenclosed sides of each decomposed box. These simplifications are limited in terms of knowledge of the object's utility, and the grasps may not afford all the abilities that grasping the object can entail and only consider the geometric information. These grasps also do not consider the task: for example, a grasp on a cup in which the fingers are inside the cup may have a good grasp quality measure but is non-optimal if we wish to pour water into the cup.

Bicchi and Kumar [2000] reviewed robotic hand design and make the case for dexterous manipulators. As has been often stated[Gaiser et al., 2008; Roa et al., 2012; Belter and Dollar, 2011], they argue that the case for anthropomorphic hands is great if a robot is to exist in the world humans have created for themselves: tools are created for humans to use and our environment is modified by humans to make it easy for us to live in, so in order to use them easily and robustly a robot must have human-like capabilities. Planning and programming actions on a kinematically complex hand is made easier when you can directly demonstrate the task as a human and assume some mapping to the robot hand. Although the cost is greater to create a hand with the required degrees of freedom to match the human hand, the need for flexibility in a robot tends to outweigh this consideration. However, high degrees of freedom can be a problem in dynamic environments using traditional methods: if the environment changes, the motion is replanned. Work has been done to alleviate this in simple, point-mass views of robots [Masehian and Katebi, 2007], but this work isn't suitable for robots with complicated kinematics.

Most of the grasping work considers simple manipulation, as in translation and rotation of objects, which is all that is needed in many cases. Other work looks at more complicated manipulations, such as the work performed by Saha and Isto [2007] that deals with the knotting of ropes. They use two simple one degree-of-freedom manipulators and manual intervention to tie knots in ropes automatically, by considering the path the end of the rope must take in winding around itself. Wakamatsu et al. [2005] also considers the problem of planning knotting and unknotting motion. They devise a notation for knots that also describes the sequence of crossings that the rope makes in performing the knot, and a set of basic operations in order to transition between states. They make a simple assumption that a grasp on a location on the rope parents that lo-

cation to the manipulator translation and rotation. A simple gripper can then produce knots on a plane. Further to this, they investigate tightening of the knot, and note that not all knots can be pulled by the ends to tighten: it depends on the topology. These complicated motions end up being generated by simple grasps and do not use in-hand manipulation.

Section Summary Grasps may be planned on arbitrary objects using concepts such as force closure and caging, but often are not dexterous, are suitable only for the final grasp pose, and require replanning when the situation changes. Most grasp metrics make the assumption that the object is rigid, any deformations in the object would require replanning the grasp. Force closure methods have been demonstrated to be a successful measure of the grasp quality. The use of force closure and caging concepts will both be considered during grasp transfer and evaluation in Chapter 5.

2.3.3.2 Grasp Transfer via Programming by Demonstration and Teleoperation

This section deals with the concept of grasp transfer in robotics, and investigates the areas of teleoperation and programming by demonstration (PbD)[18][Billard et al., 2008] in order to do so.

PbD is a concept that has become increasingly popular in robotics literature, after having being introduced about 35 years ago [Billard et al., 2008]. Traditionally, to program a robot to perform a task a human operator would have to create an algorithm that would perform both the task in a general sense and respond to any perturbations or disturbances during the task to ensure it completes successfully. This may result in manual decomposition of the task into small subtasks, and generating code for each of them. PbD allows the user to instead demonstrate the task (often multiple times) to the robot in some manner, either by controlling the robot's degrees of freedom directly or in a more abstract way, and allows the robot to abstract this task by generalising from the demonstrations. It is inspired by the technique humans employ when being taught a task by experts. Attempting to imitating these solutions to tasks allows robots to limit the search space or direct their search towards successful results. An assumption is made that these results will be successful again (if demonstrated on the same robot) or can be adapted to the novel situation in a faster or cheaper way by adapting to the new scenario (if demonstrated in a different situation or by an actor with different

[18] Also referred to as Learning from Demonstration or Imitation Learning.

morphology). This section looks specifically at PbD for grasp and manipulation transfer.

One major consideration when approaching a PbD problem is determining which concepts in the demonstration should be imitated. For example, in a manipulation task: is the hand relationship with the object important, is it only the final location of the object, or is it the overall trajectory of motion that is necessary? In a walking task, is it the joint angles (that can be directly measured) that are important, or the centre of mass staying within the support polygon, or the final location of the body? These questions can usually only be answered by the specifier of the task, and can be difficult to infer given only input data of the task.

Prior work on guiding robots through tasks physically by a demonstrator (kinaesthetic teaching) have shown promise, as they have no correspondence problem or calibration errors. Sauser et al. [2012] attempts to train the robot to resist external perturbation through initial demonstration using a dataglove, and then refinement via kinaesthetic teaching. This is limited to only grasping the trained object. Kim et al. [2014] has shown generation of robot motions for catching objects in flight. They do this by demonstration of multiple grasps on known objects by passively driving the robot's hand joints to close on the object, using the object coordinate system to store the relative configuration. A probability density representation of this grasp is created from the distribution of these relative positions and orientations. These can then be reapplied given the object configuration. However this requires many grasps to be demonstrated on known objects, and so is not suitable for transfer to novel objects or manipulators.

Although these kinaesthetic teaching approaches can result in direct training of the robot, the degrees of freedom the human uses is large compared to the demonstrated degrees of freedom, and often the interaction with the object is unnatural as the human attempts to translate their own understanding of the problem into a form that allows them to move the robot as a tool in order to achieve the task. This approach is also cumbersome for the human, as directing a robot can be much more challenging that actually performing the task using their own body.

Work on offline human grasp transfer has also been investigated. Offline grasp transfer can be considered as a specialised instance of trajectory level programming by demonstration, however the field is older than the term. Kang and Ikeuchi [1997] propose a general description of grasps that generalise to arbitrary manipulators. They classify grasps in a hierarchical way, splitting up grasps into two main classes: volar[19]

[19] Also known as 'power' grasps.

and non-volar[20] (finger tip) grasps. They use a mechanical glove along with video to capture interaction at a low frame rate of 5 frames/s. They use collision detection to adjust the hand pose as they state there are errors "...due to the both imperfect glove calibration and hand model" which they show causes incorrect information in the capture. Mapping the grasp to a robotic manipulator is performed by mapping the identified grasp style assuming the manipulator is in the same location as the human hand. They then tune the grasp by local sampling using a custom task-related criterion. This method does not consider the case that the object differs from the captured object, nor is it able to precisely place the fingers on the object, opting instead for 'pre-grasp' shapes. Pre-grasps cannot be solely used when grasping more complex objects that have special relevance to the placement of the fingers, such as a spray bottle with a trigger or a bowling ball with finger holes.

Teleoperation has been used instead to control robots in manipulation tasks with simple objects[Hertkorn et al., 2013; Grollman and Jenkins, 2007]. In these systems a joystick, motion capture glove, haptic interface or other input mechanism is used to control the actuation of the robot, whilst the human experiences the manipulation often by visual and/or haptic feedback. In contrast to PbD, this happens with direct feedback to the user, in order to achieve some task occuring at this moment, rather than for generalisation of the task for future problems. In the work of Hertkorn et al. [2013], the authors use two Kuka LWR robotic arms to provide the haptic feedback, a Cyberglove to capture the joint angles of the human hand and a virtual reality headset that allows the human to look around the environment. Visual feedback in terms of the reachable Independent Contact Regions [Roa et al., 2011] is given to aid the user in selecting a grasp. A major advantage of teleoperation is that the user can have immediate feedback as to the task and can adjust their input when the robot is failing. This utilisation of human adaptability is also a disadvantage though, as the user requires training to use the input device because the mapping to the robot is often unnatural, especially when the user is controlling a large number of degrees of freedom.

The areas of teleoperation and programming from demonstration often address the issue of grasping from the point of view of mapping the demonstrated grasp onto a robotic system with dissimilar kinematics. There are three distinct approaches in the literature:

1. **Joint-to-joint mapping** is a method that takes advantage of similarity of human grasp poses to a robotic manipulator and aims to produce grasps of similar

[20] Also known as 'precision' grasps.

quality directly from demonstrated joint angles. Dimensionality reduction methods have been used to effectively map human demonstrated poses onto different robotic manipulators [Ciocarlie et al., 2007; Do et al., 2008]. This technique has also been demonstrated on virtual humanoids with success, as there is a more direct mapping. Molet et al. [1999a] uses magnetic sensors in combination with the Cyberglove to capture the full body pose of an actor performing a task in order to transfer it to a virtual environment. They utilise the work of Boulic et al. [1996], discussed in Section 2.3.2, in order to ensure the hand does not penetrate the virtual object when closing around them, in essence power grasping when the actor closes the hand around an object. The mapping to the virtual character is joint-based, where each joint has a correspondence to the virtual character's skeleton system.

2. **Cartesian space mapping** has been used to map finger tip positions from a human to the Barrett hand [Peer et al., 2008]. These methods focus on preserving geometric relations between the two spaces and allow to transfer precision grasps. The mapping between hands with different numbers of fingers is, however, not generally well defined, and these methods do not generalise well over different object shapes. A recent extension of this approach that maps contact positions of a grasp to similar objects is presented in [Hillenbrand and Roa, 2012]. This technique warps the surface geometry of a source object to a target object along with the contact points of a grasp, but does not deal with the non-contact relationships between the hand and the object, nor does it consider the approach for the grasp. Full-body motion has been transferred using IK and optical motion capture [Ude et al., 2004] but this only transfers the broad motion, not the specifics required for interaction with the world.

3. **Pose mapping** is an indirect joint space mapping technique. In [Pao and Speeter, 1989], the authors propose a method based on functional analysis of the human hand and which results in an algebraic transformation of the human hand configuration into target domain configurations. The work presented in [Gioioso et al., 2013] shows how the pose mapping problem can be solved using a proxy object by representing the grasp by a minimal ellipsoid containing the finger tips. The inverse kinematics is then computed with respect to the shape of the ellipsoid defined by the demonstrated hand pose. Unlike the method which will be explored in this thesis in Chapter 5, this ellipsoid representation limits the motions

that can be generated via this approach.

A hybrid technique that combines these approaches has been presented by Kang and Ikeuchi [1994]. Here the authors automatically assign manipulator and observed fingers to either a primary or secondary group, and use this to determine the transfer, combining pose mapping and joint-to-joint mapping. The approach has problems with non-convex objects and dexterous interaction. Tegin et al. [2009] also used imitation learning from human demonstration to extract different grasp types, utilising pose-mapping and Cartesian space mapping. However, they do not model the whole reach-and-grasp movement and circumvent the high-dimensionality problem by using simpler manipulators. No Programming by Demonstration method has yet shown generalisation for complex, dexterous grasping and manipulation between different objects and different hand configurations.

Section Summary Grasp transfer is important in both teleoperation and programming by demonstration. Utilising human examples provides a means of performing high-quality grasps with intrinsic knowledge about objects, without having to create complex metrics directly. A difficulty is selecting a good representation such that the examples can be abstracted. Joint mapping has been demonstrated but is unintuitive for online transfer and will fail during interaction with objects for offline transfer. Abstractions of the hand, such as considering the finger tips to exist on an ellipsoid, have been used previously to transfer motion successfully in simulation but are only suitable for a subset of grasps. Chapter 4 in this thesis sets out how this limitation can be avoided and looks for an appropriate representation for a grasp and manipulation task by considering envelopment and the position of hand segments relative to the object surface. Chapter 5 then describes the method for applying demonstrations represented in this way to novel situations.

2.3.3.3 Potential-Field Guided Path Planning

Chapter 5 in this thesis sets out a method for motion transfer which uses a potential field, and for this reason the field of potential-field guided path planning demands attention.

Potential-field guided path planning constitutes a classical approach to motion planning in general. Khatib [1986] proposed to compute a virtual potential field using local areas of attraction and repulsion to guide a robot towards a goal location while

avoiding obstacles. Haex and Gambardella [1992] applied this idea to perform grid search within a robot's configuration space with a potential field to find stable grasps. They use attractive potentials to position the root of the manipulator close to the pre-selected 'focus' points or to the centre of mass of the object. Collision detection is still required as the field does not represent the surface of the object accurately. In the work of Bierbaum et al. [2009], dynamic harmonic potential fields have been used to selectively explore regions of interest when planning grasping of unknown objects. The technique sets repulsive point samples in regions that are already explored and attractive point samples elsewhere in the space, causing the robot to move to unexplored areas by following the gradient of the potential field. This allows a robot to acquire the 3-dimensional object representation from tactile data. Song and Kumar [2002] use potential fields in controlling decentralised point-based robots in the transport of simple convex objects. They use repulsive potentials generated from the point-based robots to avoid collision and an attractive potential field generated by the centroid of an object to move the robots towards the object. This combination of repulsion and attraction causes the robots to space themselves around the object whilst avoiding their neighbours.

Section Summary Each of these techniques follow the gradient of the field in order to achieve some task, be it exploration of novel space, avoidance of objects, or approach to target locations. They use discrete point charges or decomposition of the space to generate the fields. Some of the fields are harmonic, in that no local extrema occur, but non-harmonic fields have also been investigated. As sampling is used, intersection of the field with objects may occur and so this has to be considered when using them for path planning.

The work advanced in this thesis also falls into the category of potential-field guided methods, but is significantly different from previous methods. Previous techniques discussed here use potential fields with source and sink, or repulsors and attractors, to guide point-mass robots around objects and to explore novel areas in the workspace by following the gradient. This thesis is distinctive as it proposes computing an electric field surrounding the object using charge simulation, which can be utilised as an object-centric coordinate system in a manner analogous to spherical coordinates in order to guide the hand and fingers both toward and around the object, and also can be used to define a measure of coverage directly from the object surface in order to describe the envelopment of the object by the hand. A key advantage of this

approach over previous potential based methods is that these constructed field lines are non-intersecting with the object and each other, even when constructed for complex non-convex 3D objects[21]. This harmonic electric field is produced via a Boundary Element Method rather than decomposition into grid cells or tetrahedralisation of the outer space which saves significant space. This allows the entire free space surrounding the object to be parametrized, rather than a finite area. Furthermore, the field may be used not only for path planning but also for generalization of the grasping behaviour by defining the novel metric of envelopment, which is useful for transferring the grasping motion to manipulators that have dissimilar morphology.

2.4 Discussion

There has been much work done in both capture and generalisation of close interactions with objects, but no methods are able to capture motion from an actor and transfer it to novel manipulators interacting with new objects without extensive planning or human adaptation. Aesthetic considerations are rare in the robotics literature, and so for animatronics[22] little work has been done when interacting with objects. Although grasp style transfer has been studied, often the approach to the object is ignored. This changes the perception of intent of the scene, and so it is important to investigate this aspect. Further to this, sometimes non-contact relationships are important in a grasp: for example, in grasping of a spray bottle it is important for a finger to be close to the trigger in order to use the tool. With teleoperation, IK or mapping joint angles have been used, relying on the operator's adaptability to interact with the world. Investigation of alternative, object-centric relationships that encode envelopment has not been explored in great detail in either animation or robotics. Relative position and orientation between actors in scenes are usually given by skeletal relationships or centre of mass, meaning that object surface information is lost. Recently surface sampling methods have been used to define spatial relationships, but suffer from problems of sample selection when the motion has to be planned on an object. It is for these reasons that it is necessary to investigate new motion capture techniques specifically for dealing with objects, new representations of motion relative to the object, and ways of applying these to novel situations.

[21] As we do not use point samples and instead use the triangulated mesh itself.

[22] Robots used in place of actors or animated characters, often in theme parks or movies. Originally considered as a puppet animated by electronics in some manner, complex behaviour can now been seen in them such as throwing and catching.

Chapter 3

Observation: Capturing Close Interactions with Objects

3.1 Introduction

This chapter presents a novel method for motion capture whilst interacting with objects closely, with a focus on dexterous interaction. Results from previously published work are drawn upon throughout [Sandilands et al., 2012, 2013a].

Scenes of manipulating objects involving detailed hand motions are well suited to video games, but rarely used. When they are included they often have to be meticulously created by animators by hand. In films these interaction motions are more common, but still require a great deal of manual work to create. Such scenes may include characters simply carrying complex objects, using everyday objects such as pencils, screwdrivers and hammers and using kitchenware such as knives and bowls. Automatically synthesizing such movements of the human body as well as the objects is a daunting task due to the complex coordination between the objects and the actor. The fingers can slide over the surface of the object, stay rigidly with it, or avoid collisions with the object while conducting complex interactions such as wrapping and winding movements. Although previous techniques to automatically synthesize such movements have been described [ElKoura and Singh, 2003; Pollard and Zordan, 2005; Liu, 2009], they are still in the early stage of research and are not mature enough to be used in the production pipeline.

Another option, aside from automatic synthesis of this type of animation, is capture of the raw human motion via motion capture techniques. In fact, in many computer animated productions an optical motion capture system is used to capture such human

and object movements to reduce the burden on the artists. In this approach reflectors are attached to the fingers as well as the objects and the 3D movements are tracked by high resolution infra-red cameras. However, when characters have to interact closely with objects in the scene or each other, these capturing processes require a laborious post-processing stage due to the significant amount of occlusions that happen between the body and the object, as well as self-occlusions that can occur especially for the hand markers.

When synthesizing a realistic animation of close interactions, not only must the movements of the actor and objects be accurate but also the object geometry and the morphology of the actor must be represented precisely. Consider a captured motion of picking up a cup transferred to a new character: with differing bone lengths than the original motion the fingers would likely penetrate into each other and the cup as the joint angles remain the same, the distance to contact in the original motion is small, but the end-effector locations specified by these angles are different. The positions of contact are also important. An incorrectly specified object could cause significant penetrations and unnatural motion. Humans are good at estimating the physical properties of an object by sight, and unexpected motions such as these could cause a viewer to disbelieve the virtual world. For example, Burns et al. [2006] investigated proprioperception disparity versus visual penetration discomfort when a users control a virtual characters, and found that visual penetration of the character and the object was very noticeable when the hand penetrated objects: the mean distance of penetration into the object in which the users noted detection was between one and five centimetres depending on the task and viewpoint.

In order to synthesize animations of close interactions between the body and the manipulated object, this thesis takes a different approach than previous work in the field. The technique proposed uses magnetic markers and an RGB-D camera[1] to capture an actor's performance, seeking to improve the capture of close interactions by capture of both the geometry and motion; this chapter describes this and provides comparison data between character animation captured solely using an optical capture system for evaluation of the system.

There are a number of advantages to using magnetic systems for motion capture, primarily that they do not suffer from occlusion problems. A secondary advantage is that these sensors record both translation and orientation, giving six recorded degrees-

[1]This class of sensors have separate channels for colour information (red, green, blue) and depth or distance from the sensor.

of-freedom in a single magnetic marker. This allows for fewer markers than would be required for an optical system to obtain the same level of information about an actor's pose. A further advantage is that although there can be deformation of the space globally in a magnetic system, locally the relationships between points are accurate, which is most important when dealing with close interactions. Although a large amount of distortion affects the bone length constraints of the character, this can be mitigated by considering modifications to the capture environment, which is shown in Section 3.4. Practical capture results and a comparison to an optical system are presented at the end of this chapter, showing that although the global positions of joints differ between optical and magnetic captures it does not affect the intent of the motion. Also shown are example captures of various interactions including opening a jar, drawing on a paper, removing pen caps, picking up and carrying objects, and sitting on a chair. The captured data is currently published as a publicly available database [2].

Contribution The main contribution over previous work is an interaction capture technique for the hands that is devoid of occlusion issues, whilst capturing the geometry of the object and motion for rigid-body objects using an automatically computed marker-geometry offset. A publicly available database of over 200 sequences of interaction motion was created during the development of this system and is presented here as a secondary contribution (see Appendix B for a description of the captured motion). Utilising the technique and the database is necessary to reduce the need for manual creation of motions for the manipulation transfer work presented in this thesis, where interaction motions are represented in terms of the relationship with the object geometry of the actor. By making the interaction motion database publicly available, it is anticipated that the data can inspire the research community to explore close interactions with objects further, as utilising this data eases the practical considerations of acquisition of close interactions.

Disambiguation from Kry et al.'s Interaction Capture The term 'Interaction Capture' has been previously used by Kry and Pai [2006] to describe the capture of the contact forces and compliance of the hand when performing simple interactions with objects. This is different to how the term is used in this thesis: here it means the capture of the geometry and the motion of the object and the actor, without occlusion for a wide range of close interactions. This is performed in order to compute the spatial

[2] Available at `http://www.ipab.inf.ed.ac.uk/cgvu/`

relationships between the actor and the object. Throughout the thesis, unless specified otherwise, 'Interaction Capture' refers to the method described in this chapter.

3.2 Background

This section first provide a brief background of object capture and surface reconstruction techniques as this class of techniques are used in this chapter. It then explains why the magnetic system is used for motion capture in this chapter.

3.2.1 Geometry Acquisition

Geometry acquisition is the technique of digitising the geometry of an object in the world. Laser scanning, stereo-vision, time-of-flight and structured light techniques are all able to capture a 3-dimensional representation of an object. However each have drawbacks. Laser scanning is expensive and not portable. Stereo-vision systems rely on features based on colour or pattern in the scene[3] to match two images taken in different locations in order to triangulate depth. Everyday objects (and many of the captured objects in this chapter) may not have a patterned surface or enough of these features, and so these techniques cannot be used. Time-of-flight cameras measure the time taken for a pulse of (usually infra-red) light to be transmitted from the sensor, bounce off an object and return to the sensor to estimate depth. The Kinect 2, as shipping with the Xbox One uses this technology but as of June 2014 is unavailable for general purchase. Other sensors based on this technology are expensive. Finally, structured light sensors project a known pattern of light into the scene, and match this to the reference pattern to estimate the depth. This can result in noisy depth readings, but can acquire registered RGB and depth information and are both low-cost and portable. They also tend to be able to capture at high frame rates. The original Kinect sensor uses the structured light approach, and because of these reasons, this is the sensor utilised in this chapter, although the technique would work with any RGB-D sensor with a high capture rate that is capable of being moved by hand.

3.2.2 Surface Reconstruction

The goal of surface reconstruction is to take a set of samples acquired from a surface, and reconstruct the original surface. These samples can be points in space, may have

[3]For example, SIFT features [Lowe, 1999].

normals approximated from the sensor or by analytical techniques, and may have additional information such as colour per sample depending on the acquisition method. There are multiple methods of reconstructing geometry from pointclouds, such as is required in this chapter, which can be broadly separated into two categories[Berger et al., 2013]:

- **Direct Triangulation of Points** - Multiple methods that take unoriented point clouds and attempt to create interpolated surfaces by triangulation of (a subset of) the points have been proposed[Amenta and Bern, 1999; Amenta et al., 2001; Dey, 2007]. These connect the actual samples by triangles, but often have problems with noisy or missing data, as they do not extrapolate and without correct samples the correct triangles are not produced[Berger et al., 2013; Dey, 2007].

- **Implicit Surface Techniques** - A second approach is to consider the samples as a noisy approximation of the surface. These techniques create implicit functions from the input samples and triangulate an isosurface of these to reconstruct the object surface. This class of technique is generally better at dealing with missing or non-uniform data. Often distance fields are used by estimation of the plane at each point[Bittar et al., 1995] or radial basis functions created with additional virtual samples either side of the scanned points, with value based on the direction of the normal Carr et al. [2003].

A current widely used method for geometry reconstruction from noisy pointcloud data is Poisson Surface Reconstruction, which is an implicit-surfacing technique [Kazhdan et al., 2006]. This chapter makes use of this as captured RGB-D data can be noisy, but able to give approximations of the surface normal for each sample point. Because Poisson Surface Reconstruction constructs their function based on the normals of the data, and the normals are often more accurate than the locations of the points estimated by the Kinect, this technique is appropriate to use.

3.2.3 Alternative Motion Capture Methods

Visual and optical systems suffer from self-occlusion by the hand in many cases and occlusion when interacting with objects leading to manual post-processing to achieve useful data. When this occurs the result is no longer true capture of the motion but an artist's interpretation of it, and is very time consuming for the artist to fix. Systems that rely on inertial sensors such as XSens [Roetenberg et al., 2013] suffer from drift

in the sensor locations due to integration of acceleration and angular velocity measurements, meaning that disconnected components do not maintain their true relationship in capture. In close interaction with objects, this is very important and can lead to penetrations and objects 'floating away' from their true location. Mechanical systems such as datagloves have problems with calibration, must be combined with another technique to acquire the root and object location, and the end effectors accumulate any error across samples on the kinematic chain meaning that the finger tips are in the least accurate location relative to the rest of the hand [Kessler et al., 1995]. As much interaction with objects occur with the finger tips, this is a problem. Further to that, the gloves cover the palm side of the hand, which is where most interaction takes place. This can affect the captured motion. Previous work has captured full body motion using magnetic sensors (for example the work of Zordan and Hodgins [2002]), but none have captured interactions with objects and finger motion. A fuller review of the related work on motion capture for interaction is located in Chapter 2, Section 2.1.

The aim in this chapter is to introduce a framework that does not suffer from occlusions, drift or accumulated positional error by combining the magnetic sensors with the geometry data of the object captured by depth-based sensors. This framework is then used to create an 'Interaction Database' that contains a variety of everyday motions, suitable for analysis and transfer to new characters.

3.3 Method

3.3.1 Overview

A visual overview of the technique is given in Fig. 3.1. This capturing technique is composed of the following processes.

1. Visual markers are placed in the scene. The magnetic markers are placed in the scene at known 'landmark' positions relative to these visual markers and their location and orientation are recorded. This allows for computation of the transformation between the pointcloud scene (scanned visually by an RGB-D sensor) and the magnetic marker space to align the magnetic marker system to the visual coordinate system.

2. The pointcloud of the geometry of the object is then obtained via the RGB-D sensor and the surface is reconstructed. A magnetic marker is placed on each

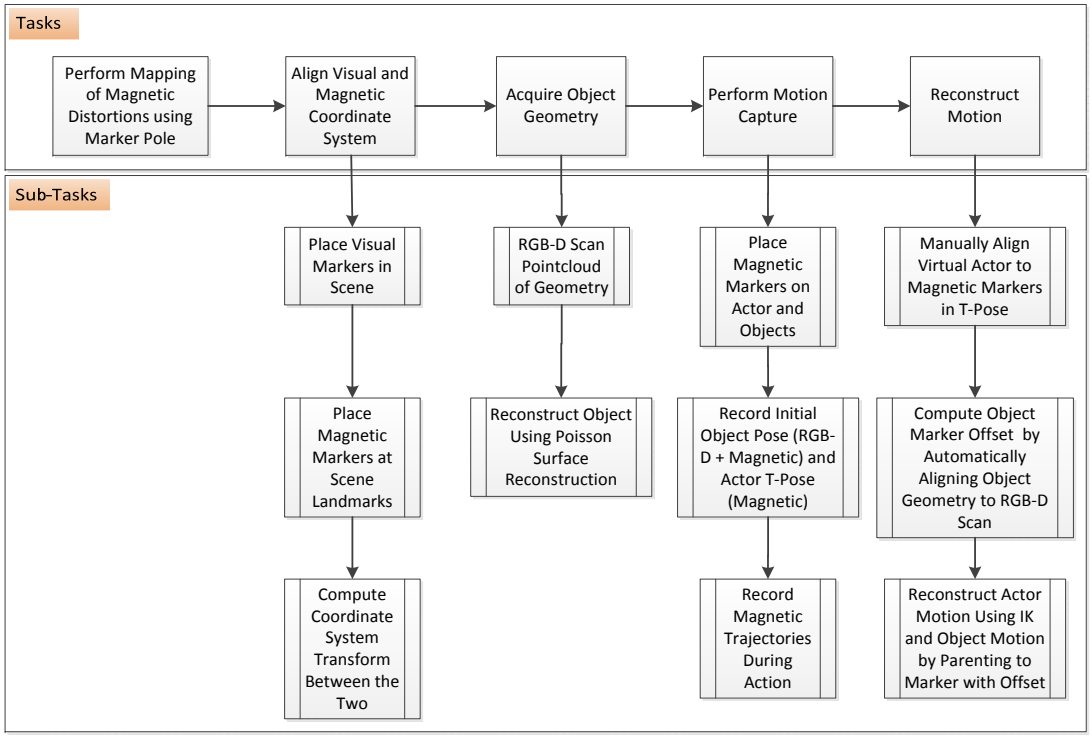


Figure 3.1: An overview of the system.

rigid section of the object and the offset of the geometry to the marker is calculated.

3. The magnetic markers are placed on the actor and the interaction is performed. The sequence of transformations of the object and actor markers is recorded.

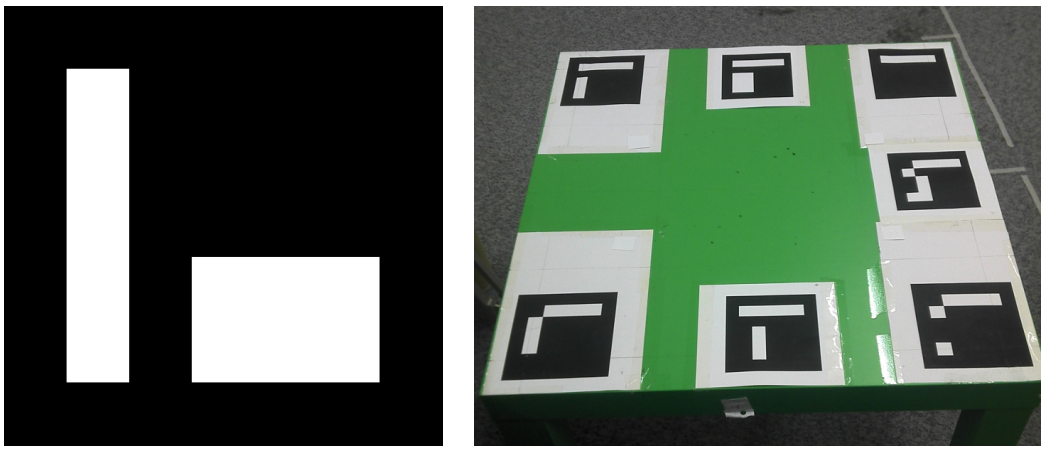
4. The relative transform of the geometry to the object is recalculated at the start of each take in case of magnetic marker slip[4] or manual change of marker location. To prevent having to reconstruct the geometry in each take, alignment of the previously reconstructed geometry to the current take is performed. This allows starting configurations that obscure some of the important geometry or would make the geometry hard to reconstruct (such as having a bottle lying down on its side), whilst retaining an acceptable quality reconstruction of the object.

5. For motion playback, the virtual actor and the object are fitted to the magnetic markers: The objects are fitted using the precomputed transformation computed in step 4, whilst the actor is fitted using the traditional T-Pose technique and bone transforms that are not captured by the markers are calculated using joint-based

[4]The markers are wired and this can cause them to move between takes.

IK.

The description of each of the processes is expanded on in the following subsections.



(a) (b)

Figure 3.2: (a) An example visual marker (b) The table with visual landmarks.

3.3.2 Aligning Magnetic Markers to the RGB-D Scene

Capture of the scene is performed using two sensor systems: the RGB-D system (in this case the Microsoft Kinect), which outputs a pointcloud representation of the geometry of the object, and the magnetic marker system, which outputs the movement of the markers (oriented points) in the scene. The main issue is that these two systems are unaligned and have different scaling, so are difficult to use together. In order to align the two sensor spaces a transformation from the magnetic marker system to the scene scanned using the RGB-D system is computed. This transformation stays consistent across takes as long as the magnetic transmitter stays static, the visual environment markers do not move and no new metal is brought into the environment.

To compute this transformation it is first necessary to define a world coordinate system that is defined by the RGB-D scan. A number of ArUco visual markers [Munoz-Salinas, 2012](see Fig. 3.2(a)) are placed in the environment in known locations. ArUco markers are detected by thresholding the image such that the boundaries are easily detectable, finding rectangles in the scene and computing the ID of each of the detected markers by considering the marker as a 6×6 binary grid. A hamming code is used on this grid to get the marker identity. Only three markers are required to obtain the location of the camera, however placing a greater number of visual markers helps when the

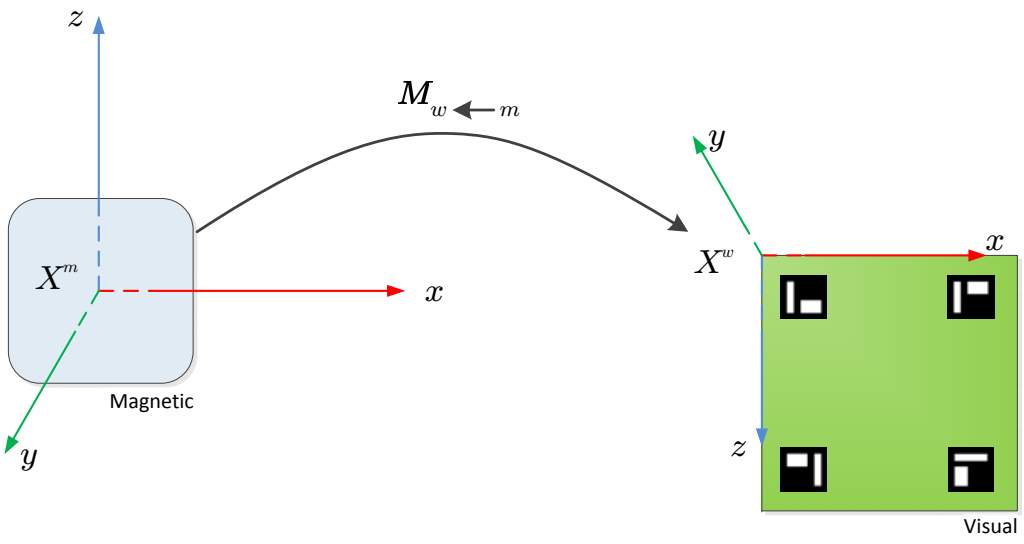


Figure 3.3: The alignment of the coordinate systems requires computing $\mathbf{M}_{w\leftarrow m}$.

object is large and may obscure some of the visual markers, and a higher number of detected markers allows for finer estimation of the Kinect camera's relative location. During the experiments in this chapter the placement of the markers is performed as shown in Figure 3.2(b).

By detecting multiple oriented visual markers in the scene an estimate of the Kinect's camera transformation can be computed and thus aligned to the world coordinate system. The magnetic markers are placed at known positions in the world coordinate system (such as the corners of the visual markers). Using the positions of the visual markers in the world coordinate system and the magnetic motion capture space, the transformation matrix to align the two environments can be obtained:

$$\mathbf{X}^w = \mathbf{M}_{w\leftarrow m}\mathbf{X}^m \tag{3.1}$$

where $\mathbf{X}^m$ is an oriented point defined in the magnetic motion capture space, $\mathbf{X}^w$ is the corresponding oriented point in the global coordinates defined by the Kinect scan, and $\mathbf{M}_{w\leftarrow m}$ is the conversion matrix between the magnetic motion capture space and the global coordinate system (illustrated in Fig. 3.3).

Computation of the conversion matrix $\mathbf{M}_{w\leftarrow m}$ is performed by combining the translation, rotation and scaling components of the magnetic marker coordinate system to the scanned RGB-D data.

The scaling vector $\mathbf{s}$ applied to the magnetic data is computed by the average difference in magnitude of the vectors joining each marker location in the world coordinate

system and the magnetic marker coordinate system:

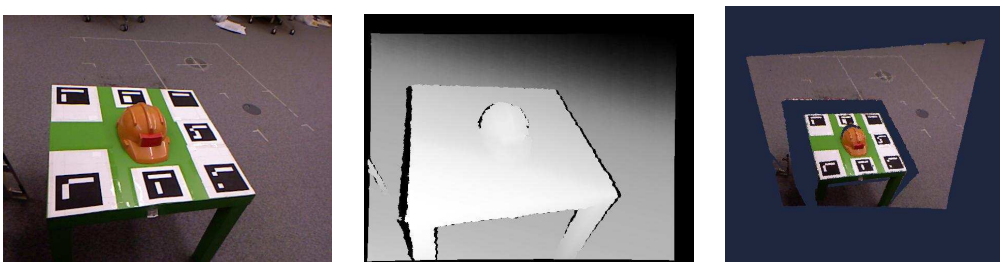
$$\mathbf{s} = \sum_{i,j \in N} \frac{||\mathbf{X}_i^w - \mathbf{X}_j^w||}{N \cdot ||\mathbf{X}_i^m - \mathbf{X}_j^m||}, where\, j > i \tag{3.2}$$

where N is the number of markers, and $\mathbf{X}_i^w$ is the location in world coordinates of the i-th marker.

The rotation correction $\mathbf{r}$ is given by the 3×3 rotation matrix computed by the average rotation around the axis orthogonal to the direction indicated by each magnetic marker and the desired direction of that marker in the world coordinate system. The orthogonal axis is obtained via the cross product of the two vectors. The angle of rotation around this axis is the angle between the two vectors computed using the dot product.

After the rotation and scaling is applied, the translation $\mathbf{t}$ is given by the average deviation in position from the marker location in world coordinates:

$$\mathbf{t} = \sum_i \frac{\mathbf{X}_i^w - \mathbf{X}_j^m}{N} \tag{3.3}$$



(a) (b) (c)

Figure 3.4: Reconstructing the geometry of an object. (a) shows the RGB data of a single frame taken from the scan. (b) shows the depth corresponding to each pixel in the RGB data. (c) shows this data combined with the camera intrinsic information to produce a coloured pointcloud for a single frame.

3.3.3 RGB-D Object Reconstruction

In order to acquire the geometry of the object, the RGB-D sensor is used to capture surface information about the object. The object is placed into the environment that contains visual markers that were processed when aligning the magnetic markers to the scene (see Figure 3.2). The sensor is slowly moved around the object at a distance of 1

Figure 3.5: Examples of reconstructed 3D objects captured by the Kinect sensor, shown next to the objects. It should be noted that there is a loss of high-frequency detail in these models, caused by noise in the original data obtained from the Kinect as well as a limited octree depth when performing Poisson Surface Reconstruction. These models can be used by an artist to design final models for display, or directly to analyse the interaction.

metre, keeping the object centred and as many visual markers in view as possible. This is performed at multiple heights around the object in the attempt to view the maximal amount of geometry without occlusion. During experiments performed in this thesis it was found that rotating around twice, first at a low elevation (approx. 30° from horizontal) and then a high elevation (approx. 70° from horizontal) worked well for most objects. By scanning the object using the sensor from multiple directions, a 3D coloured point-map of the scene is built as follows[5]. As the sensor moves around the table, the visual markers allow the sensor's transformation relative to the table to be computed. Each pixel is projected to a 3D point with colour and normal information using the depth data and the computed transformation of the Kinect for each captured frame (see Figure 3.4). Removal of points belonging to the floor plane can be done by a conditional pass filter where $z > 0$, as the visual markers are all on the $z = 0$ plane. This leaves only points belonging to the object. From this point representation of the object, recovery of the surface information is performed as follows. First, as many samples are located near to each other, downsampling the data using a high resolution voxel grid allows us to be able to process the data in significantly less time without much loss of detail as the data is not useful for high-frequency detail in any case due to noise from the sensor. This downsampling approximates the pointcloud by replacing each voxel that contains at least one point with a single point at their centroid. Performing Poisson Surface Reconstruction [Kazhdan et al., 2006] gives a triangulated mesh that closely

[5]A technique adapted from RGBDemo [Burrus, 2012]: `http://labs.manctl.com/rgbdemo/`

matches the original geometry, despite losing some high-frequency detail. Images of some of the reconstructed 3D surfaces using this method are shown in Fig. 3.5.

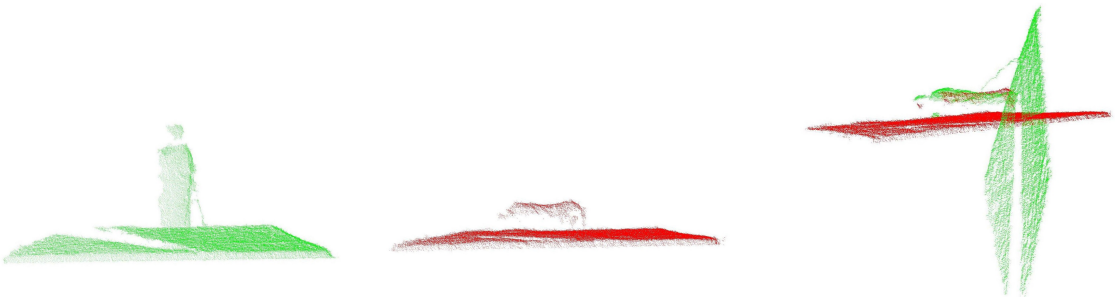
After the object is reconstructed, a magnetic marker is attached. The initial offset of the marker is then computed by the difference between the object and marker transformations.

Optionally, a further scan of the scene can be performed using the visual markers but without the interaction object in order to capture the static objects that give context to the motion. Considering that some motions may have implicit interaction with objects that do not move, such as sitting at a table, and that these static objects may be important semantically, including them in the final take is important. This scan can be performed in the same way as the capture of the object above. Due to the visual markers being on the ground surface in our system, the viewpoint on the Kinect is constrained as there needs to be multiple markers in view to calculate the Kinect's location. Because of this, reconstructing the entire geometry of the scene is not possible. However, the environment close to the markers is able to be reconstructed, and is the most important in terms of the actor's interaction.

3.3.4 Marker-Object Transform Calculation

To compute the relationship between the magnetic marker attached to the object and the object model geometry, the model and marker must be aligned. The assumption here is that the marker does not change relative position significantly during a single take. However, over a number of takes and due to interaction with the object, the marker can move relative to the object. It is also possible that the marker may have to be moved between takes in order to best capture different interactions. This renders the initial transformation between object and marker invalid, and so must be recomputed. This is done by recomputing the transformation at the beginning of each take, using the marker reading and an RGB-D scan of the scene at the first frame. In order to not have to reconstruct the full geometry of the object for each take, a single model is created for each object and aligned to the initial scan of each take. This gives a consistent model geometry across takes, but also allows for the change in relative marker position in each take.

To assist the user in aligning the object, automatic object template alignment methods of the pointcloud representation of the original object and scanned scene is used. Alignment of the object template to the new scan can be performed automatically by



(a) (b) (c)

Figure 3.6: An visual demonstration of SAC-IA alignment between the object (a) and a new scene partial scan (b) with ground plane for illustration. (a) shows the partial scan of the object to be aligned: an upright bottle. (b) shows the target scan: a bottle lying on a table. (c) shows the whole scan in (a) transformed by the computed alignment transformation matrix. The alignment is computed without regard to the ground plane.

Sample Consensus Initial Alignment (SAC-IA) [Rusu et al., 2009]. This technique takes two pointclouds and aligns them based on the local features of the pointclouds. In a preprocessing phase, the 'Fast Point Feature Histograms' are computed for both the object and the scene, which encode the relative normal and positional data locally across each point in each pointcloud. For the alignment phase, a subset of these features of the object template are selected randomly and are matched to the features in the scanned scene. The rigid transform defined by these correspondences is then applied to bring the object and scene into alignment, and a distance-based error metric is applied. After a set number of steps, the best alignment is returned as a transformation from the object to the scene, as shown in Fig. 3.6.

As the object scan does not capture the fine detail in the model, the automatic fitting does not perform well in all circumstances, especially when the object has a near-symmetric shape. In the case that the automatic fitting does not converge well, manual completion of the fitting process is performed, with the scanned data as a visual guide. Although the marker is captured during this processing, it is small compared to the object and does not tend to affect the estimation of the transformation.

Once the alignment between the pointclouds is computed the inverse of this transformation can be used to transform the object geometry to the starting configuration for the motion capture session. This gives us the transformation of the object and the object magnetic marker in the first frame of the motion take, and so computation of the

magnetic marker offset can be performed by taking the difference of these transformations.

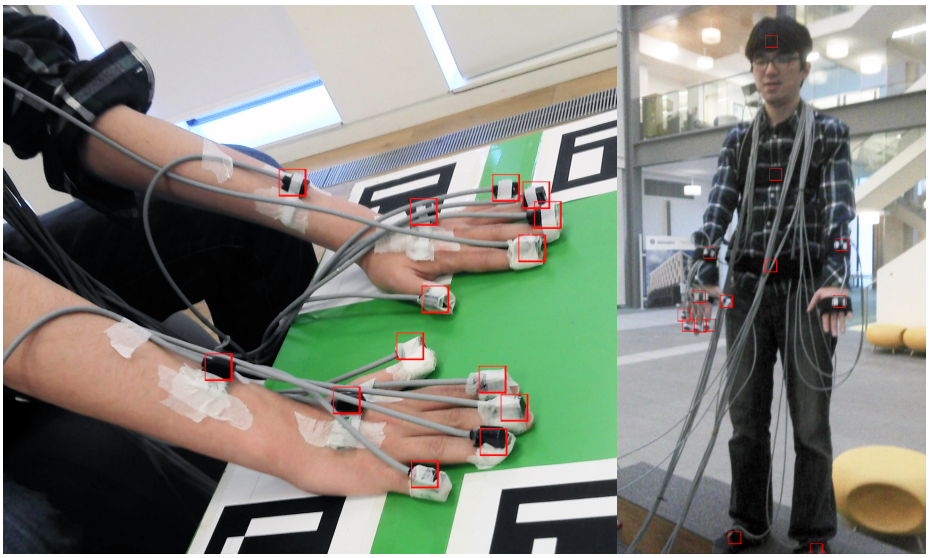


Figure 3.7: Sensors attached to the hands (left) and body (right). Their locations are marked with red squares.

Property	Value
Number of Sensors	16
Degrees of Freedom	6
Update Rate	240 Hz
Optimal Positional Accuracy	0.71 mm
Optimal Orientation Accuracy	0.15° RMS
Latency	3.5 ms

Table 3.1: Polhemus Liberty Properties

3.3.5 Capturing Motion Via Magnetic Sensors

In order to capture the human motion, magnetic markers can be used due to their accuracy and the desire for continuous data. The finger tips are often occluded by objects and the hand itself when grasping or manipulating, which reinforces the need to use a sensor that can not be occluded.

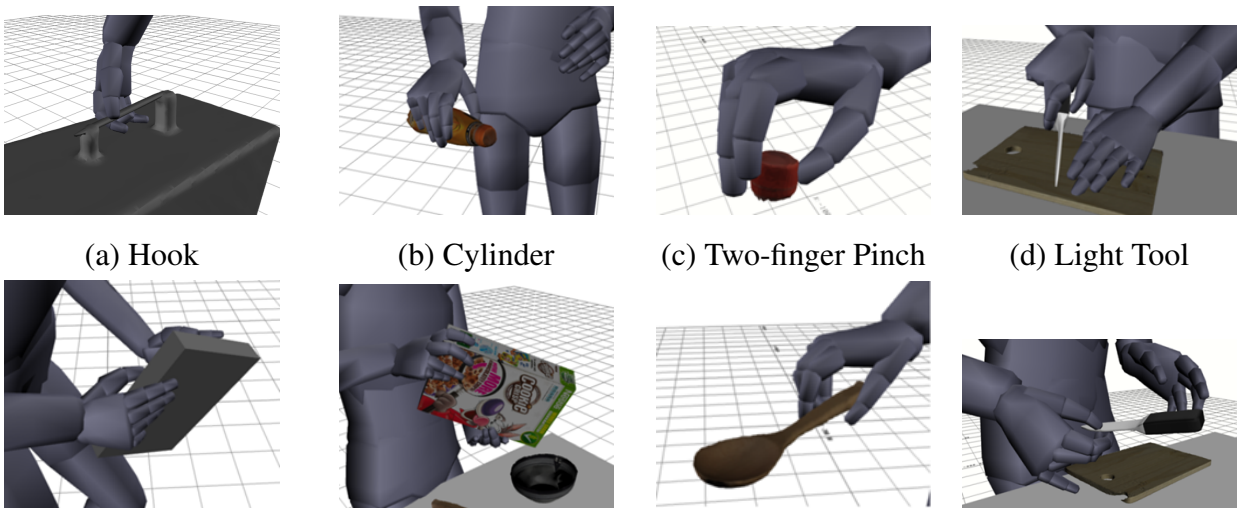
In this chapter two separate marker sets are demonstrated for capturing the human motion: one for intricate motions that require two hands, and a second set for capturing

full-body motion where only one hand requires finger capture. In the capture of intricate, two-hand interactions, a sensor is placed on the nail of each distal finger section such that positive Y is pointing from the inside to outside of the hand and positive Z is pointing in the direction of the finger, and a further sensor on the back of the hand oriented such that positive Y is facing away from the palm and positive Z is pointing in the direction of the finger base as shown in Figure 3.7, left. Further markers are placed on the left and right forearms, with positive Y pointing upwards and positive Z is facing forward of the actor when the arms are held in front of the body. When capturing full-body motion, the same number of sensors are distributed over a larger number of bones. In this case, the sensors for the right hand are placed as for the two-hand interaction configuration, but instead of the detailed left hand a single sensor on the head, chest, waist, left forearm, right forearm, left hand, left foot and right foot are placed as shown in Figure 3.7, right. For attaching of the markers, medical tape is used directly on the skin of the actor, or if that is not possible to a tight band around the actor. There is the risk of marker slip, so they are taped strongly, winding at least three times covering the marker. The cables are anchored with enough slack to allow full range of motion, but with no more than this minimum so that the cables stay close to the body and do not snag on the environment. This enables the accurate capture a single hand interaction with many objects using a limited number of sensors (see Table 3.1 for sensor details).

Both of these marker sets require a total of fourteen sensors, which allows two sensors for objects in the scene and for marking known locations. This is a hardware limitation, although in principle it is possible to use multiple transmitters and sensors as Mitobe et al. [2006] showed. This is not necessary for the majority of interaction motions though: although the number of markers for the objects are low, this is mitigated by each marker reporting both the position and orientation, allowing us to use a single marker per rigid section of the object to capture the full transformation of that section. Motion data from the magnetic system is exported as an ASCII encoded sequence of positions and orientations, tagged by sensor ID. This is then parsed by the animation system and or imported directly into Autodesk Motionbuilder by a custom plugin. As previously mentioned in the overview, some joints require IK to compute their orientation as there are fewer markers than bones. To ensure the orientation of these segments is similar to the performed motion the orientation data obtained from the magnetic sensors for the finger tips is used in addition to the positional data to further limit the solutions of the IK for the fingers, giving more accurate results than if the

positional data were to be used in isolation. In order for manipulation interactions to be accurately represented, the wrists are strongly parented to their respective markers and so can cause the forearm length to change, the exact amount of which being determined by the magnetic noise in the environment. In the worst case in the following experiments according to the data collected, this could be up to a 7cm difference (see Section 3.4.2, illustrated in Fig. 3.16), however as the direction and magnitude of error is similar for markers that are nearby, in practice it is much lower.. Although this isn't physically realistic, it ensures the areas of contact with the object are close to their true locations. This can be thought of as a hard constraint on the relative transformation between the wrist and the marker, with all other relative transformations being soft constraints.

The object motion is reconstructed by parenting the relevant marker's position and orientation using the offset previously calculated in Section 3.3.4. In this way, the object follows the change in position and orientation of the marker. In the case of articulated objects, the model is separated into rigid components and each component parented to a marker as is performed for an individual object.



(a) Hook (b) Cylinder (c) Two-finger Pinch (d) Light Tool

(e)Four-finger Prismatic (f) Two-hand Prismatic (g) Four-finger Pinch (h) Lateral Pinch

Figure 3.8: A few examples of grasp styles captured.

3.3.6 Selection of Captured Interactions

Considering which motions to capture for the database is important. In order to enhance the utility of the database, the decision was made to capture both everyday interactions with objects (e.g. drinking from a bottle, eating from a bowl, using tools),

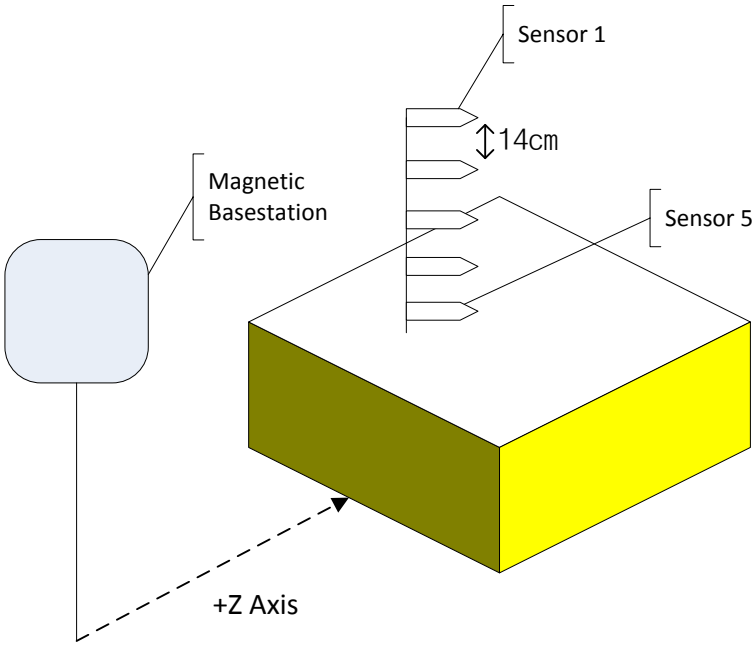


Figure 3.9: This figure shows the experimental set-up for the magnetic marker pole and basestation. The sensors are attached to a wooden pole in the same orientation with even spacing. Each sensor is 14cm below the previous one, with Sensor 5 being 1cm above the floor plane.

and motions commonly found in computer games (e.g. picking up a rifle, firing a pistol). These specific sequences were selected to ensure that a variety of different grasp styles and manipulations for both hands occurred at least once in the database as well as full body motion, such as moving a large box that requires resting it on the chest. Some of the different grasps of objects are displayed in Fig. 3.8 with their identifiers from [Cutkosky and Howe, 1990]. However, the database not only contains grasps but a range of interactions including full body motion and manipulation of objects. This means that whole sequences of motion (not simple grasp poses) are able to be analysed using this database.

3.4 Experimental Results

3.4.1 Deformation in the Magnetic Field

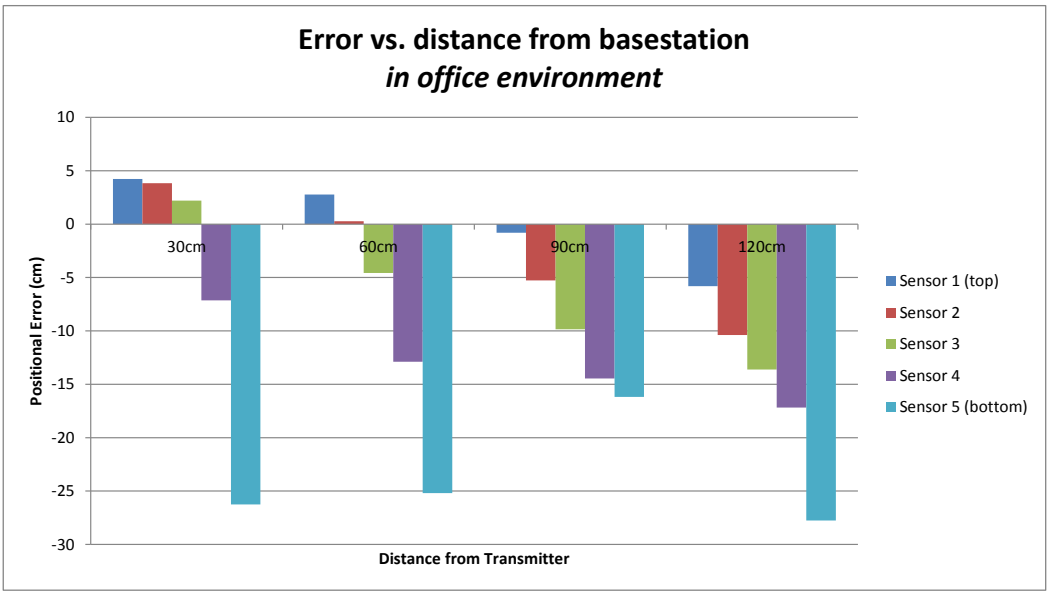
The capture-session is held in an open space to ensure less interference from conductive material. When informally testing the magnetic system in an office environment,

it was found that magnetic interference in the environment caused the accuracy of the sensor to be too low to be usable. As mentioned in the related work (Section 2.1.2), metallic objects cause eddy currents in the field, producing errors in the readings. One possible cause of error can be conductive materials in the floor affecting the magnetic field. To quantify this, an experiment was performed in a common workplace environment measuring the positional error of the markers. A 'marker pole' was set up containing five markers, each spaced at 14cm intervals in the vertical direction. This was placed at known distances from the magnetic basestation (see Fig. 3.9) and the positional error was taken. Figure 3.10 shows a comparison between the positional error in an office environment both near to the ground and on a wooden platform raised 45cm. This experiment shows that the error values in a real-world environment are somewhat larger than the theoretical optimal values stated in Table 3.1. It is also clear that the accuracy improves when a raised platform is used.

3.4.2 Distance Comparison with an Optical System

In order to compute the accuracy of the magnetic system of capture presented in this chapter, a comparison to the traditional optical approach of motion capture was performed by simultaneous capture of a set of character motions using both an optical system (the Naturalpoint Optitrack (V100:R2) 8 camera optical motion capture system[Naturalpoint Inc., 2014]) and the magnetic system. This allows for comparison of the motions of the same character captured by both the magnetic system and the optical system, by fitting a consistent human skeletal model to each marker set and comparing the joint location differences between the two systems (see Figure 3.17 for a visual comparison). Eight motion clips were captured which utilize the full range of the joints (including warm-up exercises, running and crouching) with the full-body marker set. Manual removal of segments of motion which contained occlusions or mis-labelling of markers in the optical capture was performed, along with alignment of the magnetic and optical motions temporally. The resulting total number of frames for comparison is 3206.

Optical systems are the current industry standard method of capturing individual actor motions, due to this the captured optical motion is considered as the ground truth and so alignment of the magnetic character's hips to the optical character's hips is performed. With these characters aligned it is possible to compare the relative position of each joint between the two systems by computation of the Euclidean distances between



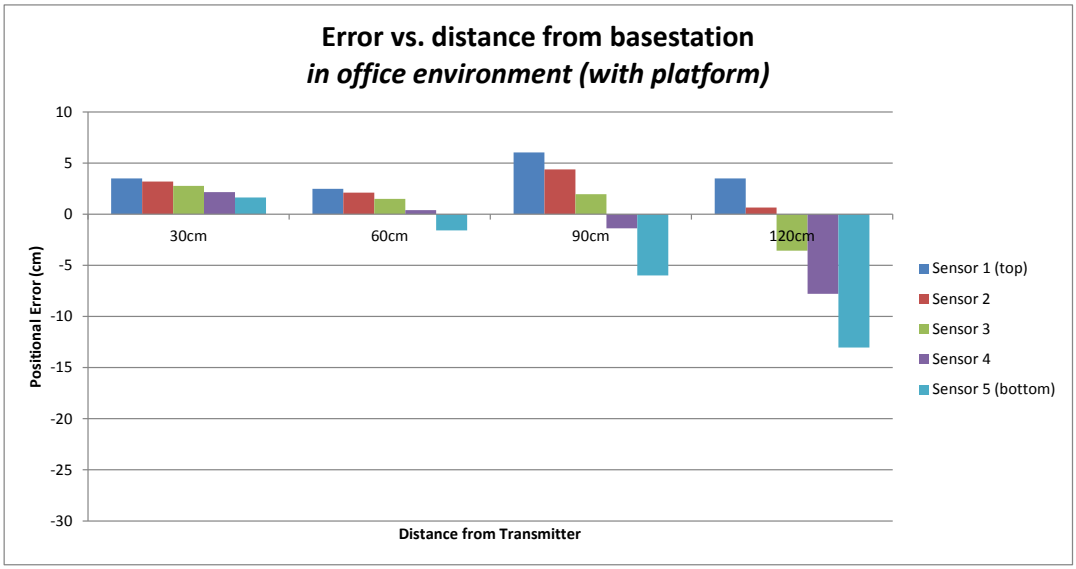


Figure 3.10: A comparison of the error in the magnetic sensor readings without and with a platform. It can be seen that a higher base (platform) for both magnetic transmitter and sensors entailed less interference and so a more precise capture environment. This data also shows the optimum distance for capture in an office-style environment is around 60cm.

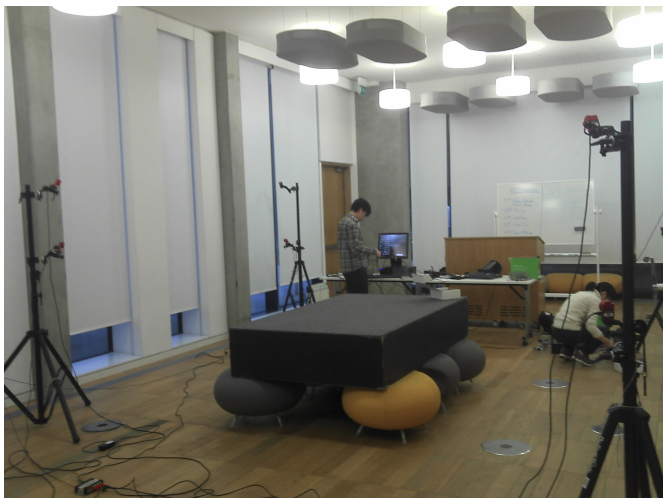
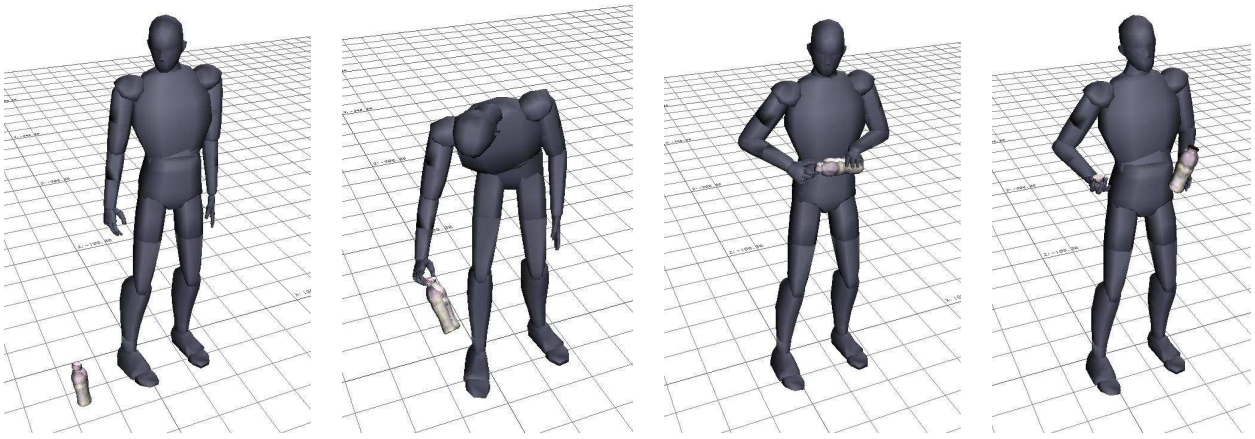


Figure 3.11: The environment for capture. The usable space is limited to the grey area in the centre (3 meters by 1.5 meters). In this image, the space is being set up for capture with both magnetic and optical sensors.

them. These results are shown in Figure 3.16.



(a) (b) (c) (d)

Figure 3.12: An example motion (bottle cap removal)

The results show that the difference in absolute joint locations between a character captured using an optical system and a magnetic system is noticeable. Some of the error can be attributed to the differences in the fitting and IK solutions when no markers were attached to sections of the body, however some of the error likely comes from the deformation in the magnetic space. As the optical system is considered to be a good estimate of the actual positions, this leads to the conclusions that this magnetic system to be less accurate in terms of absolute positions for character animation. However, the accuracy of the relative (local) positions of markers is good due to the smooth

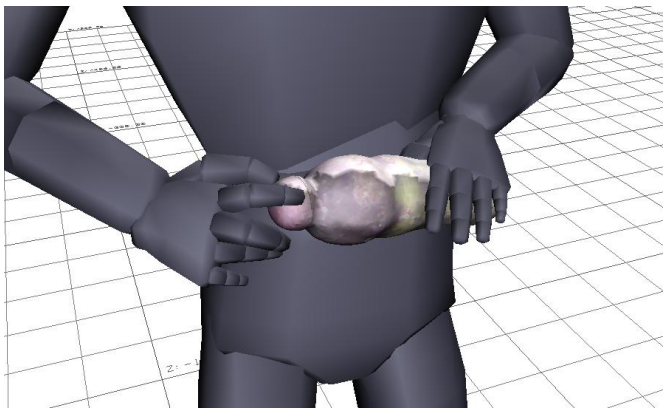
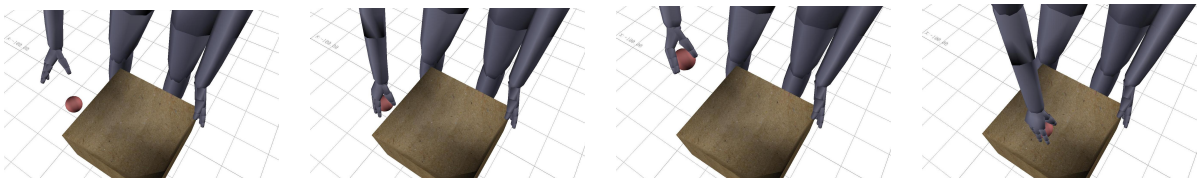


Figure 3.13: Close-up of example motion. These types of close interaction must be accurately captured to prevent serious penetration issues.

deformation of the magnetic space (illustrated in Fig. 3.10, where neighbouring sensors have similar levels and direction of error), and this is more important when capturing close interactions with objects and the environment. Visually the captured motion is semantically similar. Another outcome to note is that in general the further down the kinematic chain from the root the joint is, the greater the absolute error. This is likely due to fitting differences of the model to the original marker locations, and the greater distance from the aligned hips.



(a) (b) (c) (d)

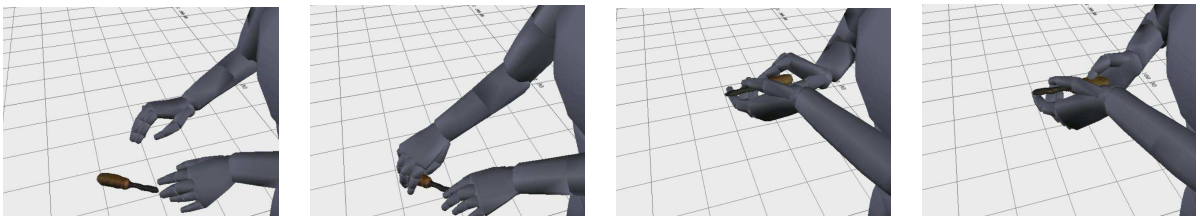
Figure 3.14: Captured motion of putting a ball into a box.

3.4.3 Capture of the Database

Taking into account these findings related to the accuracy of the sensors in the environment, the capture region for the creation of the database is set-up above the ground using wooden platforms (see Fig. 3.11) in a magnetically quiet environment. In order to compensate for magnetic field distortion in the environment a calibration is conducted in the capture space. This is done by modifying the previously constructed marker pole to space the markers at 40cm intervals (giving a maximum calibrated capture height of 2 meters) positioned at points on a 5×5 floor grid in the capture space,

and sample the magnetic position and orientation deviation from the true values. The corrective transformation is the inverse of this deviation. These are then linearly interpolated to give a continuous correction space for the markers to exist in. It is likely that the true deformation of the space is not linear, but this approach gives a useful approximation. It is notable that although the absolute position isn't always accurate, the relative location of the markers is the most important aspect for interaction with objects, and this relationship is preserved using these sensors.

Over 200 takes of different close interaction motions have been captured using this technique, with 18 different objects. These motions include using a screwdriver, picking up and putting on a hat, drawing using a pen, sweeping using a broom, drinking from different types of cup, and drinking from a bottle after unscrewing the cap. The actor body geometry is estimated using Motionbuilder, and the object and character is then fit to the markers, using the built-in IK for the character for bone segments without markers. These takes are available to download from the online database in FBX format. The FBX format was selected as the most appropriate format for multiple reasons. Firstly, this format supports both geometry information and multiple motion sequences in a single file, allowing the object, actor and optionally the environment geometry and motion to be stored in a single file along with multiple takes using each object. Secondly, the format is widely used, and can be easily imported to many graphical modelling programs, such as Autodesk Maya or 3DSMax. Thirdly, there is an SDK freely available for this format [Autodesk, Inc., 2012]. Snapshots of some of the example motions are shown in Figure 3.12-3.15. For example videos, please see the attached digital appendix and project website[6].



(a) (b) (c) (d)

Figure 3.15: Reconstructed motion of using a screwdriver.

[6] http://www.ipab.inf.ed.ac.uk/cgvu/InteractionDatabase/interactiondb.html

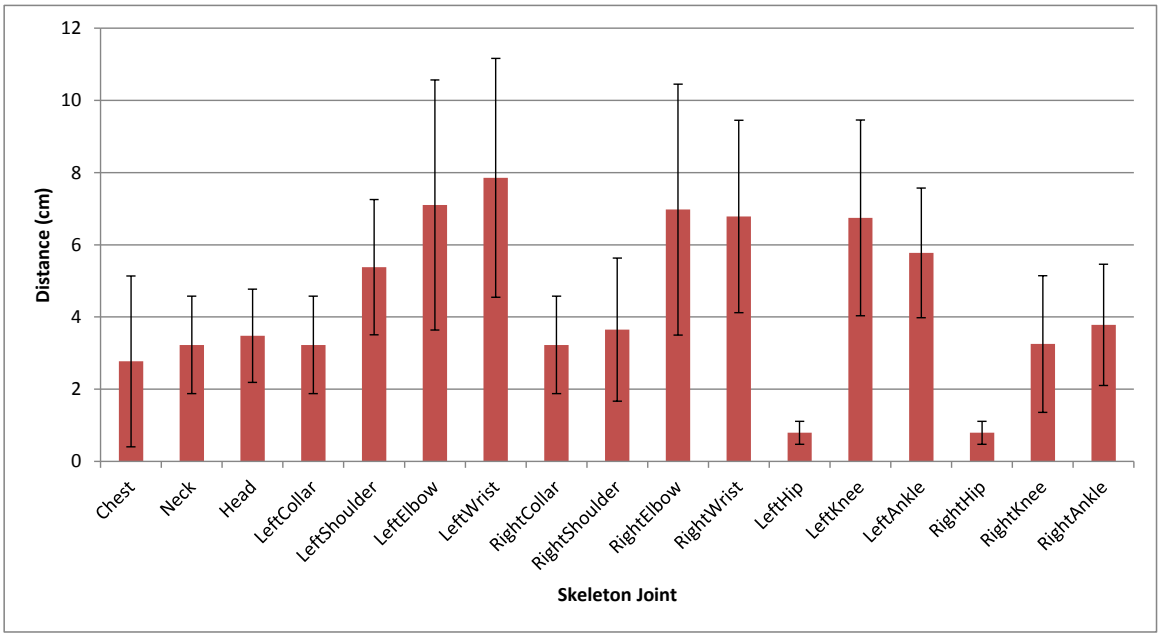


Figure 3.16: The mean distance between a character's joints calculated between optical and magnetic marker reconstructions across all 3206 frames. The error bars show the standard deviation. Although the absolute error for some joints is large, this is due to smooth deformation in the magnetic space and slight errors in the optical reconstruction, preserving the relationship between markers and limiting visual errors.

3.5 Chapter Discussion

This chapter presented a framework that can capture the human movements of manipulating objects by jointly using a magnetic motion capture system and an RGB-D sensor. It demonstrated successfully capturing interactions of manipulating a variety of everyday objects. The data is publicly available from the Edinburgh CGVU website.

Comparison with glove-based frameworks: Glove-based methods were tested as an alternative to the magnetic capture system, but these were found to have drawbacks as expected. The two glove-based motion capture systems tested [Kessler et al., 1995; Fifth Dimension Technologies, 2014] were resistance-based, in that the movement of the hand affected electrically conductive sections of the glove, changing the resistance which was then measured. This measurement infers some information about the joint

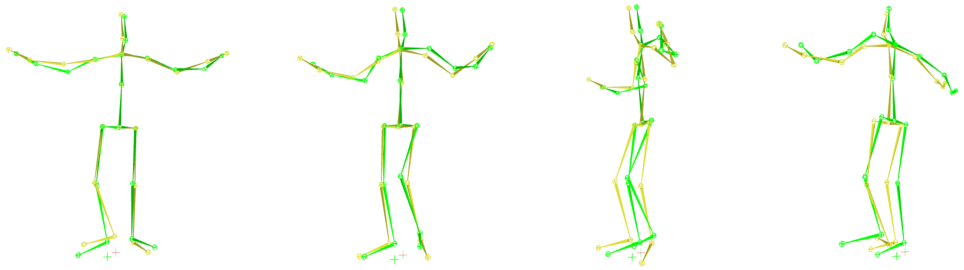


Figure 3.17: An example sequence captured using both the optical and magnetic sensors. In this sequence a character spins around clockwise to face backwards. A separate skeleton is fitted to the markers for each sensor set. The green skeleton has been fitted to the optical markers, the yellow skeleton has been fitted to the magnetic markers.

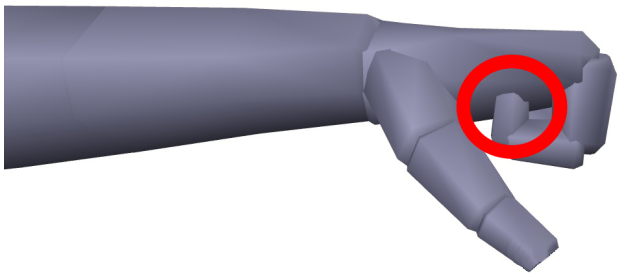


Figure 3.18: When using a glove-based system, only joint angles are recorded, so sometimes penetrations such as this one may occur due to imprecise bone lengths or error in the captured joint rotation values.

state of the hand. This requires calibration on each use, as the location and size of the hand in the glove greatly affects how the sensors are affected. This glove-based method stands in contrast to optical or magnetic sensors, as it provides the local rotation of the hand joints, greatly relying on an accurate morphology and calibration to ensure the finger tips are in the correct place rather than the absolute position and rotation which can be read from the magnetic sensors. Interaction with the environment often relies on the finger tips and incorrect positioning of them can cause penetrations, so this is a serious problem for capturing close interactions (see Figure 3.18). Futhermore, an additional motion capture system would be required for the capture of the object's motion, and for computing the wrist's global transformation. The magnetic capture system was therefore deemed superior in this case, as the markers give the absolute position and orientation of each end effector and the objects, allowing the scene to be

reconstructed.

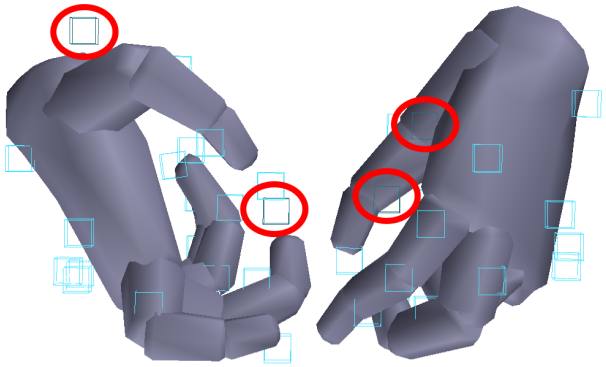


Figure 3.19: Using optical systems, occlusions may occur, even for simple manipulation motions. Here the occluded markers are shown in dark blue (circled in red), whilst the detected markers are displayed in light blue.

Comparison with optical-based frameworks: The magnetic system also appeared to be superior to an optical marker approach when dealing with close interactions. Some simple grasping motions were captured with optical markers: grabbing a cylinder from the side in an open environment. Even in this simple case occlusions occurred (Fig. 3.19). When the optical markers passed close to each other there was also the possibility of marker confusion as the finger tips are too small to place a unique set of markers to define a rigid body. These two problems meant that many takes of captured data required interpolation between optical markers and manual cleanup of the data. It also should be noted that although the precision of the magnetic system is lower than the optical system in general, the relative position and orientation of markers is accurate, and this is what makes this system suitable for capturing close interactions. The advantage the optical system has, however, is that the markers are small and wireless, meaning motions that require entirely uninhibited motion, such as throwing a ball, may be captured successfully if occlusions can be prevented.

Drawbacks: Although there are benefits to using the magnetic system, there are also problems to overcome. As the markers attached to the actor are also the sensors, they require a method to transmit the data to the base station. This means using a system that has a power supply for each marker and a wireless transmitter (increasing the size of each marker and decreasing the accuracy) or using wired markers as the system used in this chapter. In these experiments the wires constrain the capture volume to around 3

metres $\times$ 2 metres $\times$ 2 metres, and have the possibility of affecting the actor's motion in some capture situations. When motion over a large area needs to be captured, this system would not be appropriate. As metallic items significantly affect the magnetic field, it should also be noted that interaction with metal objects cannot be captured using this technique.

The geometry capture technique also has drawbacks. The depth noise in the Kinect sensor can be on the order of centimetres [Khoshelham and Elberink, 2012] and this produces inaccurate pointclouds, meaning that the geometry with lose detail in a similar range of error. As technology progresses, the error due to hardware will decrease and this technique shall yield improved results. Even with these low-frequency detail reconstructions of objects, the interaction is visually clear and can be analysed using this data.

Taking the work further: The system described in this chapter is limited to capturing objects that can be treated as rigid or articulated bodies. This limitation means that interaction with truly deformable objects such as ropes and clothing cannot be captured in this way. Future work may be able to combine the magnetic sensors with the Kinect to capture the changing geometry during the take, which may then be able to be registered to each other in a deformable manner (rather than rigid registration) to produce the motion of the deformable object as well as its geometry. Such an approach may also allow greater precision in reconstructing the object's motion during takes.

Chapter 4

Representation: Envelopment and an Object-centric Coordinate System

4.1 Introduction

This chapter discusses the parametrisation of the space around an object to simplify the description of motion relative to the object (specifically considering the approach, grasping and simple manipulation motions of objects). In doing so it introduces a novel metric for envelopment of a triangulated object by an arbitrary triangle mesh, as well as introducing a novel object-centric coordinate system based on concepts from electrostatics. It draws upon my previously published work, in particular [Sandilands et al., 2013b] and [Wang et al., 2013a][1].

Spatial relationships are important in interactions, as they determine the artistic style and type of interaction in many cases[Nakaoka and Komura, 2012]. Wrapping and envelopment motions are particularly important in certain types of motion; they are particularly pertinent in grasping as the hand often envelops the object. Existing research in the animation and robotics literature has not identified relationship invariants that capture the continuous concept of envelopment in three dimensions (although non-continuous measures of coverage have been proposed such as Igarashi and Suzuki [2011]'s interactive cover design system). The concept of envelopment is useful in grasping and therefore there is a need to compute this property. Two techniques for this concept of measuring envelopment have been developed over the course of this

[1]The initial work on defining the Electric Parameters in this section was performed in collaboration with Kirill Sidorov and He Wang, where I took the primary role in defining the electric coordinate system. As clarification, the implementation, experiments, extensions, and analysis in this chapter are my own work.

work: Winding Numbers and Electric Flux. Each have their own benefits and drawbacks which will be examined in this chapter.

The relative location and orientation of points is also useful in defining interaction. Accordingly, attention has been paid to developing a technique that avoids the limitations of the Interaction Mesh and similar point-based approaches to spatial relationship description whilst retaining the benefits of motion represented by the spatial relationship.

Chapter Outline Winding numbers are the basic measure of the amount a curve in two dimensions envelops a point. This chapter starts by describing an extension to the winding numbers that quickly computes surface 'envelopment' of a triangulated object in three dimensions. Experiments are shown using this envelopment metric for power grasping, and an explanation about why this has problems for arbitrary shapes follows. An extension to the Interaction Mesh which defines the relationship between a character or robot's skeleton and the geometry of an object is then described and discussed. However, using the concepts relating to the Interaction Mesh with objects has drawbacks under certain circumstances, for example in generation of novel motion or with sharp changes in object geometry. It is, therefore, necessary to consider other relationship representations and so the Electric Parameters (which combines the two concepts of envelopment and relative configuration in a single representation with greater computational complexity but more general application) are presented here with a particular regard to object interaction. Discussion of other applications of this representation in the field of animation and graphics is presented as well in order to show the general utility of a representation such as this one.

The Electric Parameters are shown to create an object-centric coordinate system and surface parametrisation that is capable of representation of motion based on the object geometry directly. Further to this, a general envelopment metric of arbitrary objects can be created from this representation. Although the Electric Parametrisation represents both the envelopment and relative configuration of a character or agent to an object, it is more computationally complex to compute than the winding number or Interaction Mesh extensions. However the Electric Parameters are applicable to arbitrary three-dimensional objects and are more descriptive in terms of spatial relationship, as they describe a continuous envelopment representation for arbitrary objects that hasn't been discussed in the fields of animation and robotics previously, so the increase in computation time is understandable. That being said, this chapter looks to

both parallelization and intelligent refinement of objects in order to reduce computation time.

The chapter finally proposes a dense correspondence technique that works between objects of arbitrary genus, which is necessary for transfer of motion to novel objects, utilising previous methods and proxy correspondences generated via the electric field. A method for detection of singularities in the field, which may allow for path planning that passes through topological tunnels in objects, is also outlined.

Contributions

- Experiments using an extension to the Winding Numbers for computing coverage of a sphere in three dimensions to power grasp convex objects.
- A set of techniques to accurately and quickly compute the electric parameters for triangulated objects.
- Definition of an object-centric coordinate system and envelopment metric that is computed directly from the object surface rather than sampled, and is valid in the entire outer space.
- Introduction of a dense correspondence technique that works between objects of arbitrary genus.

4.2 Winding Numbers and the Concept of Envelopment

In mathematics the winding number[2] is the metric for the amount a point is surrounded by an oriented closed curve in the anti-clockwise direction in two dimensions. This can be calculated by discretising the curve into linear segments and computing the sum of the angle between the start and end points of each of these segments:

$$w = \frac{1}{2\pi} \sum_{i=1}^{N} \frac{\arccos\left[(S_i - P) \cdot (T_i - P)\right]}{|S_i - P||T_i - P|} \tag{4.1}$$

where w is the winding number, N is the total number of line segments, and S_i and T_i are the start and end points (respectively) for the i-th line segment (Fig. 4.1 shows this for a single segment). Importantly this considers the direction of winding, with anti-clockwise being represented by a positive winding value, and a clockwise winding being represented as negative winding.

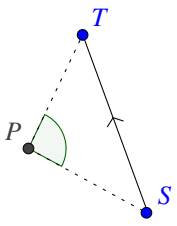
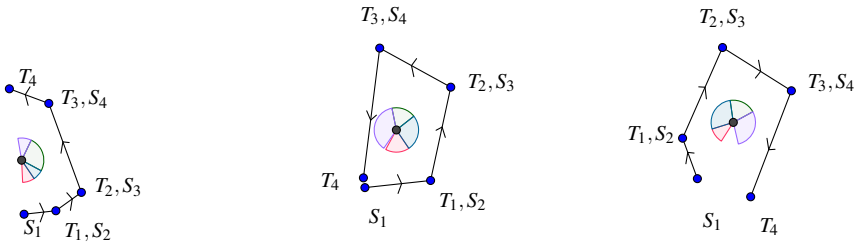


Figure 4.1: Computation of the winding number in two dimensions for a single line segment $\vec{ST}$ around a point P is performed by simply computing $\frac{\angle SPT}{2\pi}$.



(a) Winding Value: ~ 0.5 (b) Winding Value: ~ 1 (c) Winding Value: ~ -0.8

Figure 4.2: Examples of how the winding number varies in different configurations in 2D. Note in (c) the reversed direction of the line sequence.

This computation allows for describing the wrapping or envelopment of a curve around a point in two dimensions even when the curve is open. Examples can be seen in Fig. 4.2.

This can be extended into three dimensions by considering the projection of triangulated surfaces onto a sphere. A single triangle can be projected onto a unit sphere by normalising the vector between the vertices of the triangle and the centre of the sphere (see Fig. 4.3). This creates a spherical triangle whose surface area can be computed using its angles α, β, and γ:

$$A = \alpha + \beta + \gamma - \pi \tag{4.2}$$

By taking the ratio of the projected triangles' areas to the overall area of the unit sphere a measure of envelopment C of a surface over a point or sphere is computed:

$$C = \frac{\sum_{i=1}^{N} A_i}{4\pi} \tag{4.3}$$

To preserve the directionality of the envelopment measure, the dot product of the triangle normal and the vector between the sphere centroid and the triangle centroid

[2] Also known as the contour winding number.

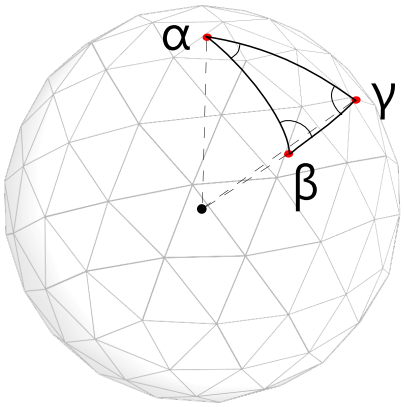


Figure 4.3: The spherical triangle has angles α, β, γ which define the area on a unit sphere by subtracting the total sum of them from π.

is considered. When they are facing in opposite directions (i.e. the dot product is negative) the triangle can be considered to be facing the sphere and the resulting area is positive. When this is not the case and the dot product is zero or positive, the resulting area value is negated. This allows for a smooth change in C as the triangle rotates from positive, to zero, to negative coverage based on the relative orientation of the triangle to the sphere.

This is somewhat similar to the concept of the solid angle, which measures the area of an arbitrary object onto a unit sphere. The solid angle does not take into account orientation however, and so is not useful as an envelopment measure for a surface that has an 'inside' and an 'outside', such as the hand's surface. This is why the three-dimensional winding number is used instead of the solid angle in this thesis.

Although this property allows a representation of the envelopment of a point or sphere by a triangle mesh, it does not directly define coverage over an arbitrary mesh. Direct projection onto an arbitrary mesh is possible by projecting toward the centroid of the mesh, however this can have unusual results when the object is long and thin, or unusually shaped. Consider a horseshoe where the centroid is outside the object: projecting onto this centroid may give a zero coverage value for many configurations, yet this is not what is intuitively understood as envelopment. Instead, manual intervention can be used to select a point inside the object with which the coverage should be computed. However this requires manual selection per object and so is not ideal.

Experiments using this technique for control of a hand during power grasps shows that plausible approach and grasping motions can be quickly computed for simple objects. Fig. 4.5 shows entirely automatic grasping from arbitrary poses by local gradient

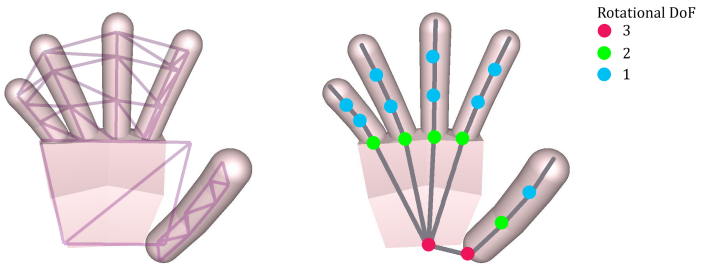


Figure 4.4: Left: The triangles used to compute the C value for a hand model. Right: The rotational degrees of freedom for each joint. The wrist also has 3 degrees of freedom for translation.

ascent of the value C computed over a triangular mesh that covers the inside of the hand. This is performed by representing the hand as a skeleton with the triangulated surface representing the enveloping surface of the hand constructed from the skeleton nodes and points with fixed offsets from them (Fig. 4.4). The triangles are oriented such that the C value with reference to a point when the hand is open and the palm of the hand is facing it is positive (the normals point outwards from the the the palm). The Jacobian of C with respect to the variation of each of the degrees of freedom of the hand skeleton is created via the finite difference method and inverted using a weighted pseudoinverse [Klein and Huang, 1983]. As the coverage is increased both by moving the hand toward the object, and by spreading the fingers around the object, the hand approaches and envelops the object from arbitrary initial configurations. Collision is prevented with the object by contact constraints added to the Jacobian. This representation has the aforementioned problems regarding objects that significantly deviate from spheres and so is not suitable for all types of grasping, but is useful in the limited domain where real-time computation is needed for simple objects. These successful examples also show that grasping by envelopment maximisation is plausible in certain circumstances. The topic of envelopment is therefore returned to in Section 4.4 with the creation of a metric for envelopment of arbitrary object geometry given a surface.

4.3 Interaction Mesh Concept Extension to Objects

Techniques that aim to preserve the spatial relationship between characters and simple objects have been developed by considering the weighted relative vectors between skeleton nodes in the scene. The Interaction Mesh as defined by Ho et al. [2010] connects the ‘nearby’ skeleton nodes of two characters using Delaunay Triangulation in

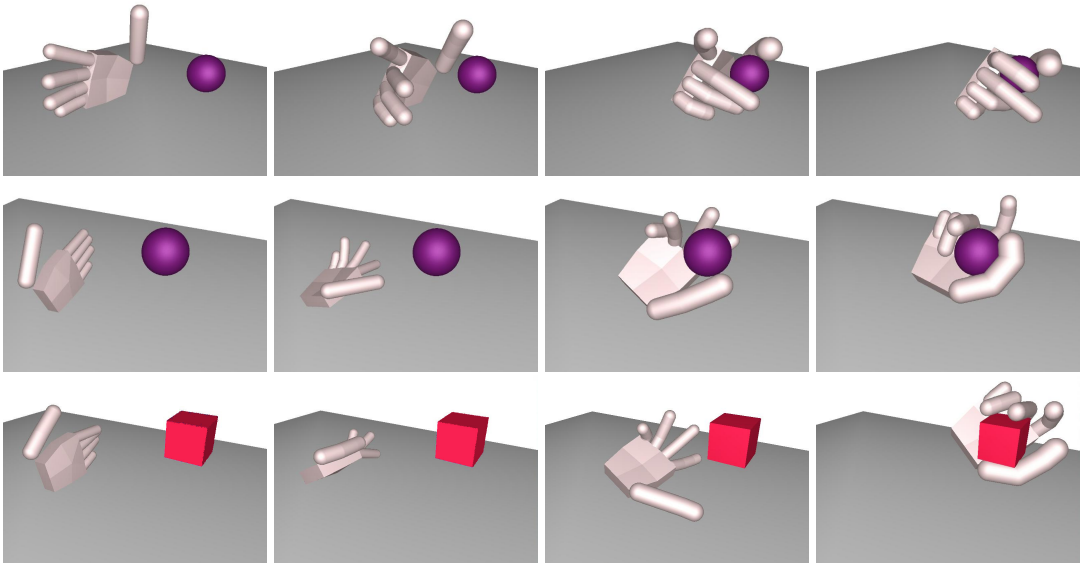


Figure 4.5: Examples of power grasping simple objects from different starting postures solely using the coverage measure C.

a graph-like structure and attempts to preserve the Laplacian coordinates of the connections when the character configuration or bone length changes. These Laplacian coordinates are defined as:

$$L(\vec{p}) = \vec{p} - \sum_{l=1}^{N} w_l \vec{p}_l \tag{4.4}$$

where $L(\vec{p})$ is the Laplacian coordinate of point $\vec{p}$, N are the number of neighbours of $\vec{p}$, $\vec{p}_l$ are the neighbouring points and w are the normalized weights of the neighbours, which are set as inversely proportional to the distance between $\vec{p}$ and the neighbour $\vec{p}_l$. This weighting prioritises relationships that are nearby. Note that this is just the relative vector from $\vec{p}$ to the weighted average of the neighbouring vertices.

This has previously been used for defining interactions between characters by minimisation of the deformation energy of the mesh:

$$E(\mathbf{V}) = \frac{1}{2} \sum_{\vec{p}_j \in \mathbf{V}} ||\delta(\vec{p}_j) - L(\vec{p}_j)||^2 \tag{4.5}$$

where E is the deformation energy, $\mathbf{V}$ is the location of the skeleton nodes, and $\delta(\vec{p}_j)$ is the original recorded Laplacian coordinate before deformation for bone node $\vec{p}_j$.

This technique has been used not only for character-to-character interaction but also for interaction with simple objects, although it requires manual creation of skeletons for these objects. These skeletons do not consider the surface of the object, and

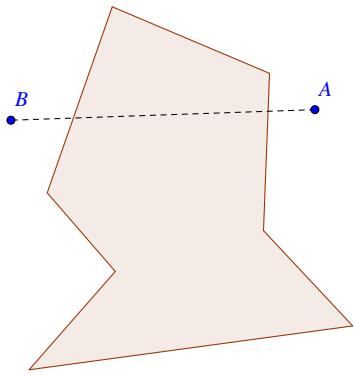


Figure 4.6: When using target positions A and B alongside Euclidean distance as the interpolation metric, collisions occur. Collision detection and response can push the interpolated point out of the object, but this often results in unnatural motion.

so when the dimensions of the surface change or contacts occur further from skeleton nodes, penetrations and unnatural movement can occur. With the types of close manipulation motion considered in this thesis, penetrations are indeed likely to occur.

An extension to deal with this problem is to consider samples on the triangulated object surface generated from proximity to joints in the human skeleton. By uniformly sampling across the surface there are a set of potential contact locations. As the human interacts with the object, the closest N surface sample points in a radius r from each skeleton node are recorded and their relative vectors are stored, alongside triangle ID and barycentric coordinates. If the object surface of model is transformed, the deformation energy E can then be preserved in the same way as before, by attempting to minimise the deviation of the relative vectors. However, these samples are not a good general representation for interaction motions as they cannot be used in a generative fashion; they can only play back the pre-recorded motion.

A further drawback to the relative vectors technique is that they do not consider the rotation of the object. In the way they are defined using the Laplacian coordinates, even if the rest of the scene rotates, the relative vectors do not: this is not helpful in most scenes. Recently Al-Asqhar et al. [2013] have developed this idea further, extending the concept of spatial relationship encoding using relative vectors with the use of the relative orientation step, given by the normal, tangent and co-tangent on the object surface and the frame of reference of the human skeleton. However, this approach is sample-based and still has problems in the generation of new motion or the transfer

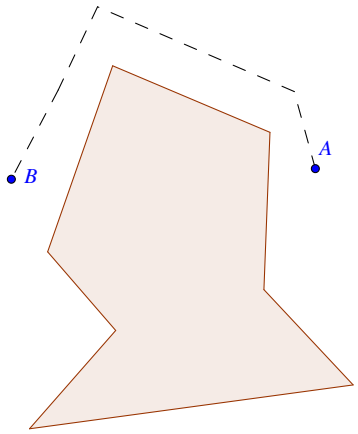


Figure 4.7: The desired interpolation path between the two locations A and B.

to novel manipulators when the correspondences between joints is not obvious. As a Euclidean distance metric is used for distance between poses, when the target posture is obscured by an intermediate invalid posture (e.g. when a bone must pass through the object to get to the target posture) there is no direct way to locally interpolate the poses that do not result in penetration (Fig. 4.6). This can be alleviated somewhat by additional sampling, but there is then a further problem of choosing the number and location of the additional samples. Furthermore, adapting the motion to new situations can be difficult as the sample locations are fixed on the surface. Physical properties such as grasp quality cannot be optimised as relationship to the true surface of the object is not encoded. It is not clear how to produce novel contacts in new locations using this representation, and as such, is only suitable for motion playback.

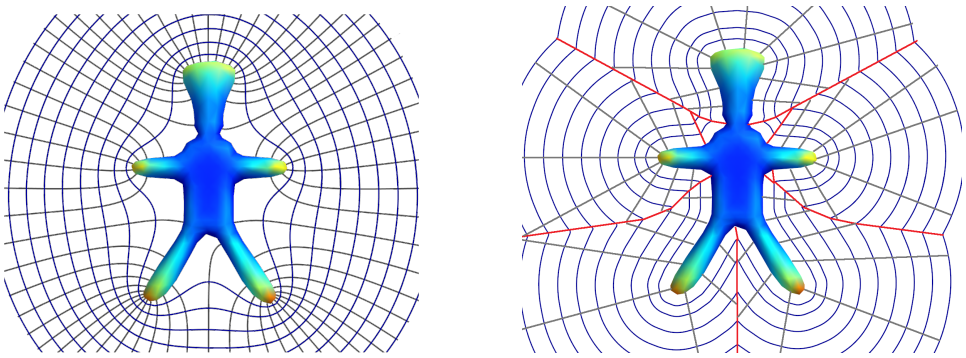
With the intention of addressing these identified problems Section 4.4 defines a relative coordinate system based on object surfaces that can handle object affine transformation and transfer to new objects seamlessly. This relative coordinate system also allows planning new contacts on the surface as the contact space is continuous rather than sampled, thus enabling local linear interpolation in the novel coordinate system such that penetrations do not occur with the object, such as in Fig. 4.7. This parametrisation of the outer space does not require choosing samples or choosing how much to sample the object, removing some aspects of parameter tweaking that may otherwise be required using alternative techniques.

4.4 Electrostatics for Envelopment and Relative Configuration Encoding

Generating and adapting spatial relationships of interactions with objects requires more than a surface-sample based method or dealing only with simple classes of object. Instead, this chapter introduces an object-centric coordinate system, which allows the spatial relationship between the joints and the object to be extracted. This, therefore, allows for motion generation relative to all areas on the object rather than only relative to sampled locations. This relationship with the overall shape, rather than just with discrete samples upon it, ensures the ability to plan motion without collision using local techniques, and measure the envelopment of the object as well as relative locations and orientation of points. This object-centric coordinate system also allows for interpolation of relationship in a way that does not cause interpenetration of the character or penetration into the object.

In this thesis, concepts from electrostatics are the tools used to parametrise the open space around an object in order to define the relationship between a manipulator (such as the human hand or robot) and the object to be manipulated. This parametrisation is created by simulating the object as a virtually charged electrical conductor in order to define an object-centric curvilinear coordinate system surrounding it. The relative position and orientation of points being controlled with regard to the object (such as a hand or manipulator) are defined using this coordinate system, describing the spatial relationship of an approach and grasp to an object. A measure of envelopment of a surface around an object can also be defined in the same representation using the electric flux: controlling a hand using this parameter allows the user to automatically approach and wrap the fingers of the manipulator around an object. A deformation in the object shape smoothly deforms the coordinate system, allowing for motion transfer to novel objects.

This section defines the terms used in describing the electric field (Section 4.4.1) and the properties that these fields have that makes them suitable for spatial relationship representation (Section 4.4.2). It then proceeds to define the parametrisation used that describes the relationship (Section 4.4.3), and finally discusses a technique for accurately defining the surface charge distribution required for this method (Section 4.4.4).



(a) Electric Field (b) Distance Field

Figure 4.8: A comparison of the isosurfaces and field lines of the electric field (left) as compared to the distance field (right). The red lines in the distance field example show example regions in space that project to multiple points on the object using the distance measure. The model colour shows the surface charge densities using the Jet colour scheme, with red being high and blue, low.

4.4.1 Electric Field Definitions

In physics, an *electric field* is a field generated by a set of electrically charged particles which describes the force on a second (positively charged) particle due to the first set of particles[3]. An electrically charged particle in the field has a scalar value defining the magnitude of the electric field called *electrical potential*, which is measured in the units of *volts*.

The rate of flow of the electric field passing through an area is defined as the *electric flux*. It is computed as the surface integral of the normal component of the electric field passing through a surface.

When considering a charged object, the *charge density* is defined as the measure of charge per unit area on the object's surface. With knowledge of the object's surface and its charge density, the potential and field in the outer space of the object can be computed[Tipler and Mosca, 2007].

[3] An electric field also describes electrical forces generated by a time-varying magnetic field, the discussion in this thesis will be confined to the first definition as the source of our field is a virtually charged object.

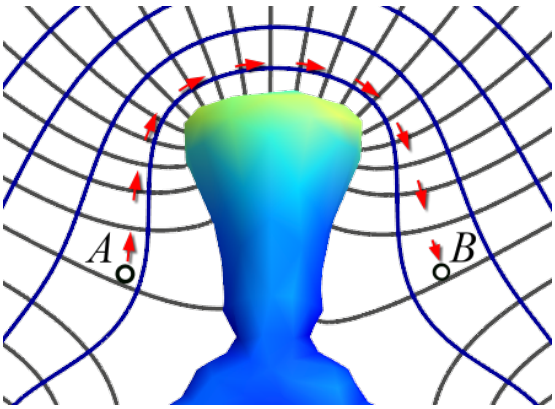


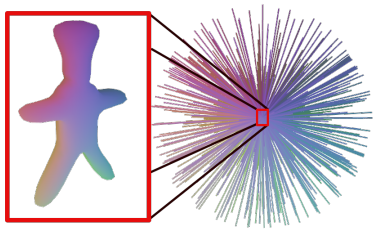
Figure 4.9: Interpolation between point A and B can be performed directly in the electric coordinates without penetration with the object.

4.4.2 Why Use The Electric Field for Motion Planning?

The electric field around a statically charged object has several interesting properties that make it suitable for use in control of particles and surfaces in the outer space of an object. Unlike the distance field (Fig. 4.8b), the electric potential around a charged object is a harmonic function, meaning that there are no local extrema in the outer space of the object regardless of the object geometry. At the surface of the object the potential is maximal, and the potential smoothly drops off to zero as the distance from the object's surface approaches infinity. At the object's surface the negative gradient of the potential is in the direction of the surface normal, as can be seen in Fig. 4.8a. As the field is harmonic the electric field lines (paths following the negative gradient of the potential) do not intersect, guaranteeing a path to the surface for any outer point by locally following the positive field gradient. This also means that the field diverges to points at infinite distance from the object.

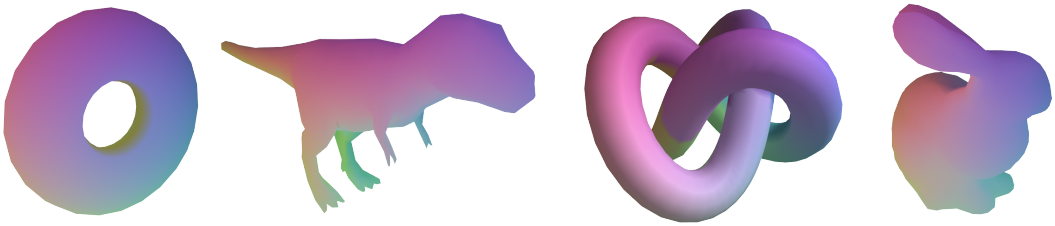
As a result of these properties, the electric field lines can be used to define a continuous mapping from an object to a sphere at infinity without flipping of triangles. It is therefore possible to define a two-dimensional parametrisation over the surface by computing the elevation and azimuth of each point traced to a near-zero potential(Fig. 4.10). When there are topological tunnels[4] there are seams in the mapping, but these exist only in the concave areas of the tunnel (Fig. 4.11). Dealing with the location of a point passing 'through' a seam requires special handling of distance across the surface as explained in the next chapter (Section 5.3.1), but in many cases there are no seams and spherical geodesics can be used. The outer space can then be defined in

[4]In this thesis the term 'topological tunnels' is used as used by Dey et al. [2013].



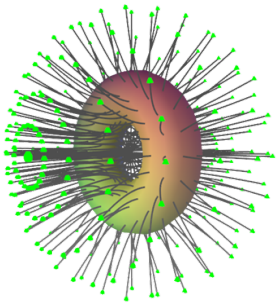
(a) Field lines traced from the model.

(b) Parametrisation shown on sphere.



(c) Further examples.

Figure 4.10: The two dimensional parametrisation of the surface is computed by tracing the field lines and taking the spherical coordinates at the endpoints.



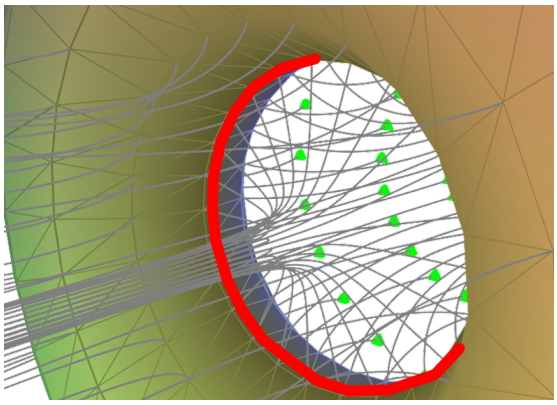


Figure 4.11: Left: The field lines traced to a fixed potential from a torus (grey with green endpoints). Right: An illustration of the seam (red) in an object with genus. The torus has a seam running around its inside. The field lines traced near this seam diverge.

an object-centric curvilinear coordinate system analogous to spherical coordinates, in that the potential at a point defines the distance measure along a field line from the object and the projected elevation and azimuth values can be used in the same way as in the spherical coordinate system. This coordinate system can be used to linearly interpolate between points in the outer space of the object without collision using spherical linear interpolation (SLERP), as shown by the example in Fig. 4.9. Deformation of the object also deforms the coordinate system, allowing relative location to be defined between objects in the electric coordinates.

The electric flux of the field also holds an interesting property: according to Gauss' law, a closed surface surrounding a charged object will always have a constant flux value, no matter the relative transformation or deformation of the outer surface (Fig. 4.12b):

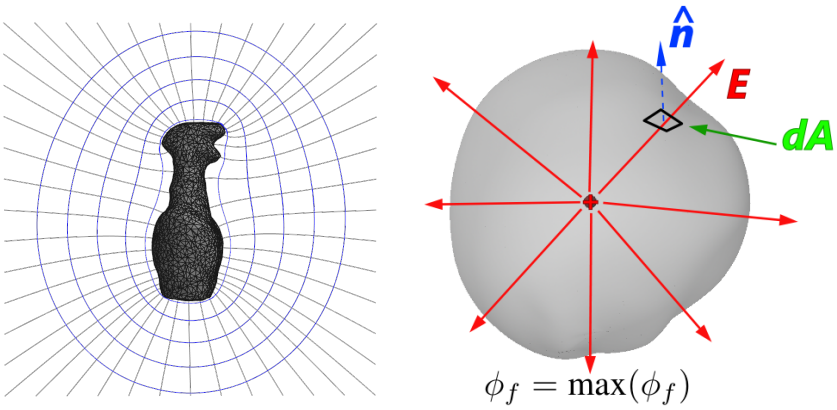
$$\oint_S \vec{E} \cdot \mathrm{d}\vec{A} = \oint_S \vec{E} \cdot \vec{n} \mathrm{d}A = \frac{Q}{\varepsilon_0} = \text{const}, \tag{4.6}$$

where $\vec{E}$ is the electric field being integrated over the surface S surrounding the charged object (with charge Q). $\mathrm{d}\vec{A}$ is a small region of S (a vector with magnitude $\mathrm{d}A$ pointing in the normal direction $\vec{n}$), and ε_0 is the electric constant. This means that the total electric flux through a surface depends only on the charge enclosed by that surface, and not on the shape of the object. For a non-closed surface, the flux varies with envelopment of the object as the integral does not account for the whole field. This only holds for electrostatic-style fields whose strength drops off at $1/r^2$ for 3-dimensional space[5]. This feature makes the flux suitable for representing the coverage of the hand or manipulator around an object.

Advantages of the Electric Field representation over other Fields Other potential fields can be generated that have non-intersecting field lines aside from the electric field. However, the electric field is superior for use in this thesis for the following reasons:

- Flux: The envelopment of a surface around an object can be directly computed from the object's surface using the electric flux.

[5] In two-dimensions the length of a line two times as far from the object surface is twice as long, so the strength must drop off at $1/r$. For three-dimensional space, the area at twice the distance is r^2 where r is the distance moved, so the field must drop off at $1/r^2$. The general requirement for Gauss' law to hold for a field is that the potential falls-off at exactly $1/r^{D-1}$ where D is the dimensionality of the space[Tipler and Mosca, 2007].



(a) Electric Coordinates (b) Electric Flux

Figure 4.12: The two components of the electric parametrisation. (a) A two-dimensional slice of the field lines (black) and isosurfaces (blue) of the virtual electric field on a scanned spray bottle. (b) A positive electric charge surrounded by a surface. The surface has a constant electrical flux no matter its deformation or geometry as long as it remains surrounding the charge.

- Non-discretised object and space: The electric field can be computed for a triangulated object directly using a Boundary-Element Method (BEM), meaning it is not necessary to discretise either the outer space of the object or the object itself.

- Defined over the entire external space: As a consequence of the BEM one can define the field everywhere external to the object without large storage requirements, and compute the exact potential given the object surface for any point rather than interpolation of samples.

4.4.3 Electric Parametrisation

Now it is clear that the electric field could be useful for object-centric control, the parametrisation must be defined. This thesis defines *Electric Parametrisation* as the abstract and intuitive parametrisation of the open space that describes the spatial relationship between a virtually charged object and a set of points and/or meshes. It specifically encodes the relative location of points in space by the object-centric curvilinear coordinate system which is called the *Electric Coordinates*, and the coverage or envelopment of a mesh by the *electric flux*. The Electric Coordinates are uniquely defined by three values: potential V, projected azimuth ψ, and projected elevation θ,

similar to the spherical coordinate system. V can be computed directly from the object surface, whilst ψ and θ are computed through tracing the field lines. The combination of ψ and θ for each vertex gives a two-dimensional parametrisation of the mesh itself that we call UV coordinates in reference to the two-dimensional parametrisations used in computer graphics for texture mapping and other operations (with ψ as the U coordinate, and θ as the V coordinate).

A two-dimensional illustration of the Electric Coordinates is shown in Fig. 4.12a where the electric field lines are shown in black and the isolines (isosurfaces in three-dimensions) are shown in blue[6]. A comparison with the distance field is shown in Fig. 4.8, showing that the Electric Coordinates have the advantage of continuous projection to the surface (the projected point does not jump to new areas as the source point moves in space) and a non-ambiguous projection. The motion of a point perpendicular to the potential field lines can be described by the projected azimuth and elevation, whilst keeping the potential constant. Conversely, motion toward and away from the object can be defined by varying the potential whilst keeping the projected azimuth and elevation constant. This allows for decomposition of motion towards the object and around the object.

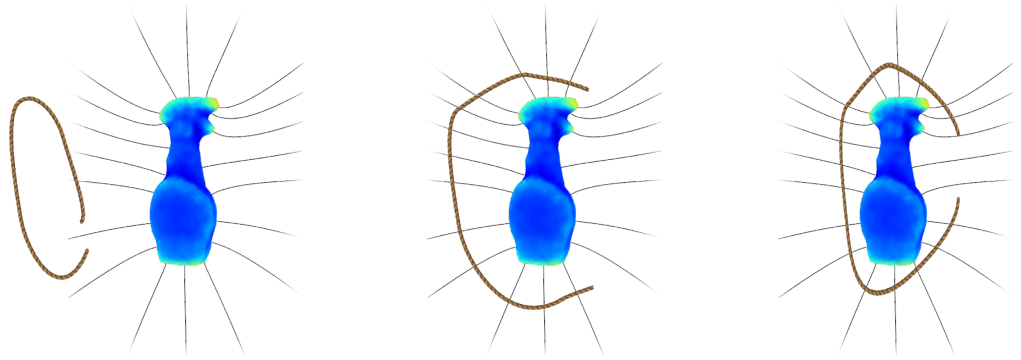


Figure 4.13: A 2 dimensional illustration of different configurations of a deformable object and a charged reference object labelled with the corresponding flux value of the deformable object.

The second component of the Electric Parametrisation of the space outside the object is the electric flux, which computes the envelopment of a surface around the object (Fig. 4.12b). This can be used to describe how a surface can move towards an object and wrap it, as shown in the two-dimensional example in Fig. 4.13. In contrast

[6]The isolines and isosurfaces of the electric field are the regions for a certain potential which have constant electric potential and vary only in ψ and θ.

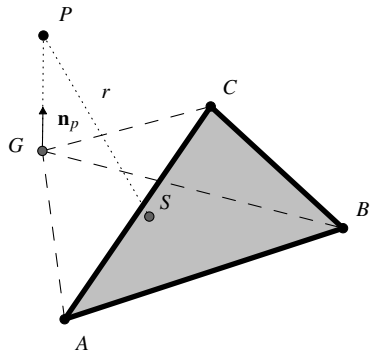


Figure 4.14: An illustration of a point P, charged triangle Δ_Q (in grey) and the geometric concepts used for electric potential calculation.

to the extension to the winding number outlined in Section 4.2, this envelopment metric is valid for any arbitrary object topology and shape, rather than only being suitable for convex, genus zero objects.

4.4.3.1 Computation of the Electric Field and Flux

In order to compute the electric field we consider a virtually charged object whose surface is represented by a closed triangular mesh T_{obj} of L triangles with per-triangle charge densities $\mathbf{Q}_{\text{obj}} \in \mathbb{R}^L$, a point P or set of points in the outer space of the object (that are attached to the manipulator) and a flux measurement surface T_m that is a mesh of N_m triangles (also attached to the manipulator).

Electric Potential In this thesis, computing electric potential analytically requires a uniformly charged triangle Δ_Q with charge Q and a test point P for which the potential V is calculated. S is defined as the uniformly charged surface of the triangle Δ_Q and $r = |\vec{PS}|$ as the distance from point P to the triangle surface.

The potential V is computed as:

$$V = \int_{\Delta_Q} \frac{1}{r} dS \tag{4.7}$$

As in the work of Goto et al. [1992], this is computed by the sum of integrals over three sub-triangles, which can be defined as a function $V(P, \Delta_{ABC})$. Full details of this function are covered in Appendix A.

For multiple triangles, the potential is simply the sum of individually computed potentials multiplied by the per-triangle charge density Q (as defined by the superposition

principle):

$$V(P, T_{\text{obj}}, \mathbf{Q}_{\text{obj}}) = \sum_{i=1}^{N_{\text{obj}}} V(P, \Delta^i_{\text{obj}}) Q^i_{\text{obj}} \tag{4.8}$$

Electric Field The electric field for a single triangle is defined as the negative gradient of the electric potential. This can be analytically computed for multiple triangles as the differentiated form of (4.8), as shown in Appendix Section A.

Electric Flux The flux is the measurement of the electric field through a surface. In this thesis the charged surface is defined by a closed mesh of triangles representing an object's surface, and the surface through which the flux is measured (henceforth called the *flux measurement surface*) is an arbitrary mesh of triangles that may be unconnected, partially connected, open or closed and consists of one or more triangles.

Due to the superposition principle, overall flux Φ_{Flux} is computed via a computation that occurs between each individual triangle representing the object ($\Delta_{\text{obj}} \in T_{\text{obj}}$) and each individual triangle in the flux measurement surface ($\Delta_m \in T_{\text{m}}$), along with the surface charges computed for the object $\mathbf{Q}_{\text{obj}}$.

$$\Phi_{\text{Flux}}(T_m, T_{\text{obj}}, \mathbf{Q}_{\text{obj}}) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_{\text{obj}}} f_{\text{flux}}(\Delta^i_m, \Delta^j_{\text{obj}}) Q^j_{\text{obj}} \tag{4.9}$$

where f_{flux} is approximated by four point Gaussian quadrature of the object triangle as follows:

$$\begin{aligned} f_{\text{flux}}(\Delta_{ABC}, \Delta_{DEF}) = \frac{|\vec{DE} \times \vec{DF}|}{2} [&\phi(\Delta_{ABC}, x_1) + \phi(\Delta_{ABC}, x_2) \\ &+ \phi(\Delta_{ABC}, x_3) + \phi(\Delta_{ABC}, x_4)] \end{aligned} \tag{4.10}$$

where

$$\begin{aligned} x_1 = \frac{4D+E+F}{6}, x_2 = \frac{D+4E+F}{6}, \\ x_3 = \frac{D+E+4F}{6}, x_4 = \frac{D+E+F}{3} \end{aligned}$$

To compute the flux through a triangle Δ_{ABC} given point charge P an analytical solution has been defined by Van Oosterom and Strackee [1983]. Let $\mathbf{a} = \vec{AP}$, $\mathbf{b} = \vec{BP}$, $\mathbf{c} = \vec{CP}$. Finally, the flux $\phi(\Delta_{\text{ABC}}, P)$ is computed via

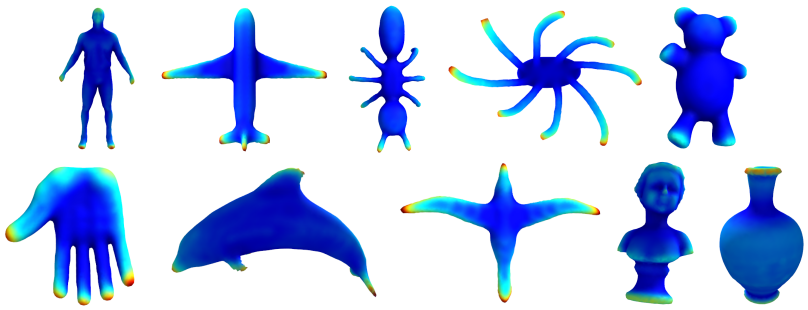


Figure 4.15: Surface charge density display on the surface of a variety of models. Red/blue is high/low charge density respectively.

$$\tan\left(\frac{\phi(\Delta_{ABC},P)}{2}\right) = \frac{(\mathbf{a}\times\mathbf{b})\cdot\mathbf{c}}{|\mathbf{a}||\mathbf{b}||\mathbf{c}|+(\mathbf{a}\cdot\mathbf{b})|\mathbf{c}|+(\mathbf{a}\cdot\mathbf{c})|\mathbf{b}|+(\mathbf{b}\cdot\mathbf{c})|\mathbf{a}|} \tag{4.11}$$

Time Complexity The time complexity of computing the potential is $O(nm)$ where n is the number of sample points for which to compute and m is the number of triangles in the mesh. Similarly, the flux computation time complexity is $O(lm)$ where l is the number of triangles in the flux measurement surface. These computations can be practically sped-up by computing the potential and flux in parallel for each triangle-point pair, which is shown in Section 4.5.

4.4.3.2 Surface Charge Simulation

The potential field is computed using a Boundary Element Method approach as described above, which requires the object surface to have the correct charge density $\mathbf{Q}_{obj}$ for each triangle, which must be first computed. The distribution of charge on the object is such that the surface is at equilibrium (the gradient of the field in the tangent direction to the object's surface is zero). If this were not the case, the field would cause the charge distribution to change as the discrete charges would move in the tangent direction of the field across the surface (in the physical interpretation of the electric field). After this computation the object is considered electrostatically charged. Two techniques to perform surface charge simulation are shown in this section. Example charge distributions on a variety of models are shown in Fig. 4.15.

This section first discusses the concept with the simpler case of matrix pseudoinverse to compute the charges. This has $O(L^2)$ space complexity, and so can be a problem for solving for meshes with a high number of triangles. To address this, the Robin Hood technique of charge simulation, as first described by Formaggio et al. [2011], is

considered.

Matrix Pseudoinverse In order to compute the surface charges by matrix pseudoinverse, a dense linear system of L equations is constructed, each representing the notion that the potential must be 1 volt at a set of probe points on the surface of the object, with L number of variables denoting the unknown total charges $\mathbf{Q}_{\text{obj}}$ for each of the triangles:

$$\begin{cases} V(\vec{x}_1, T_1, Q_1) + V(\vec{x}_1, T_2, Q_2) + \cdots + V(\vec{x}_1, T_L, Q_L) = 1 \\ \vdots \\ V(\vec{x}_L, T_1, Q_1) + V(\vec{x}_L, T_2, Q_2) + \cdots + V(\vec{x}_L, T_L, Q_L) = 1, \end{cases} \tag{4.12}$$

where $V(\vec{x}_m, T_l, Q_l)$ denotes the potential V at point $\vec{x}_m$ due to the l-th triangle carrying charge Q_l in the analytical form as defined in Equation 4.8. For the probe points $\vec{x}_m$ the barycentres of the mesh triangles are selected (although any L non-coincident points on the surface of the object would be suitable). Equation 4.12 defines a dense linear system that can be written down in matrix notation as $\mathbf{PQ} = \mathbf{1}$ by noting that $V(\vec{x}_m, T_l, Q_l)$ can be written as $V(\vec{x}_m, T_l, 1)Q_l$. This system is typically ill-conditioned, and therefore it is solved using the pseudo-inverse of $\mathbf{P}$ in the least squares sense.

This technique has some drawbacks: it requires constructing a $L \times L$ matrix that stores the effect of each triangle on the potentials at the barycentres at each triangle. As the space complexity is $O(L^2)$, this prevents solving for meshes with large numbers of triangles. The time complexity is $O(L^2)$ for creation of the dense linear system and $O(L^3)$ for solving it. Although in order to compute the field this charge simulation is only required to be performed once for each model, this is slow for larger models. It is notable that although the matrix is dense, each row is dominated by only a few terms which are the potential values of the triangles nearby to itself. The Robin Hood method addresses these drawbacks by numerically solving by iteratively updating the charges until the potential at the surface is close to one for all triangles.

The Robin Hood Method The Robin Hood method is an approach to charge simulation that uses only $O(L)$ space whilst remaining fast ($O(L^\alpha)$ where $\alpha < 2$ [Formaggio et al., 2011]). The concept is simply to iteratively move charge from areas where the potential is high to areas where the potential is low. By swapping the required amount of charge, the lower and higher potential areas are brought into electrical equilibrium (their potentials become equal). Formaggio et al. [2011] showed that there is no effect

of critical slowing down of this method, even at very high precisions for a variety of randomly generated surfaces, meaning that although there is no formal proof of convergence the practical evidence points to convergence happening in all cases. Their experiments involve remeshing surfaces into right-angled triangles to speed-up computation, but non-right angled meshes also do not experience critical slowing down. This is backed by the experiments performed for this thesis, in which every model tried converged to a solution, including animated models.

The following algorithm (Algorithm 1) describes how this computation can be performed practically.

Input: Object mesh T_{obj}, convergence threshold ε

Output: Charge density per triangle Q_{obj} across the mesh such that the surface has constant potential.

Initialise $\mathbf{P}, \mathbf{C}, Q_h, Q_l$;

forall the $q \in \mathbf{Q}_{obj}$ **do**

    q = rand(0,1);

end

forall the $\Delta \in T_{obj}$ **do**

    Compute potential at barycentre of triangle Δ with unit charges and store in P_i;

end

repeat

    Compute the current potential for all triangles given our charges by $P_i \cdot Q^i_{\text{obj}}$ and store in C_i;

    Find highest and lowest potential in C_i and store in Q_h and Q_l respectively;

    if $Q_h \parallel Q_l$ *are outside one volt* $\pm\varepsilon$ **then**

        Transfer the charge from Q_h to Q_l such that the potential of each triangle is one volt, updating Q_{obj} in the process;

    end

until *Highest potential* Q_h *and lowest potential* Q_l *are both within one volt* $\pm\varepsilon$;

Algorithm 1: Robin Hood iterative method of charge simulation.

This algorithm increases linearly in time taken as related to the number of triangles in the mesh as shown in Fig. 4.16. Although charge simulation is a comparatively time-consuming process, it only has to be performed once per static object.

For animated meshes however, this iterative method is additionally beneficial because although the charges must be redistributed when the mesh deforms, it can be assumed that the spatial differences in the mesh are small as the time between frames

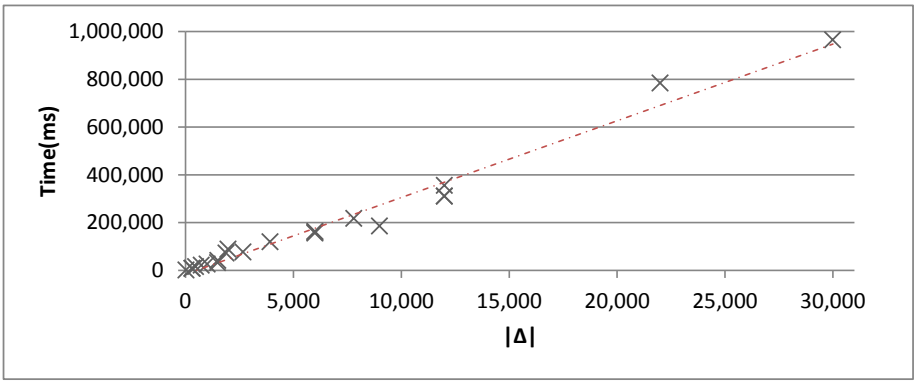


Figure 4.16: The time taken to compute the charge densities across the surface of a variety of meshes in the database from scratch. These 21 meshes vary from consisting of 20 triangles up to 30,000. The relation between time to compute the charge density and number of triangles is approximately linear, as displayed by the red trend line.

Figure 4.17: Examples frames from the run cycle of a human displaying the surface charge densities by colour as in previous figures. High charge densities (red) can be seen in very concave areas such as the bent elbow and the inside of bent knee joints.

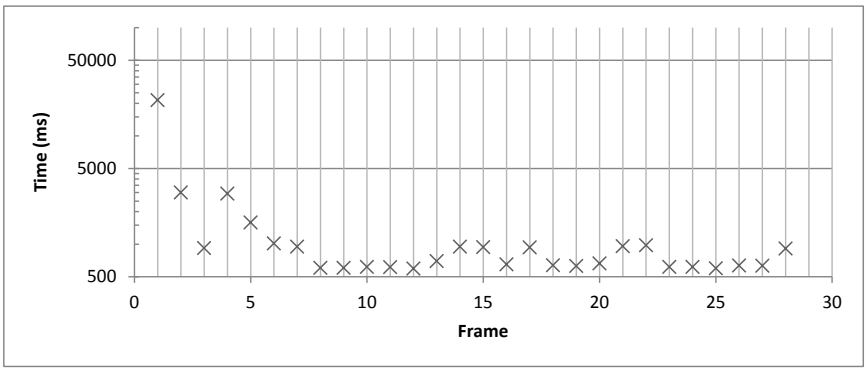
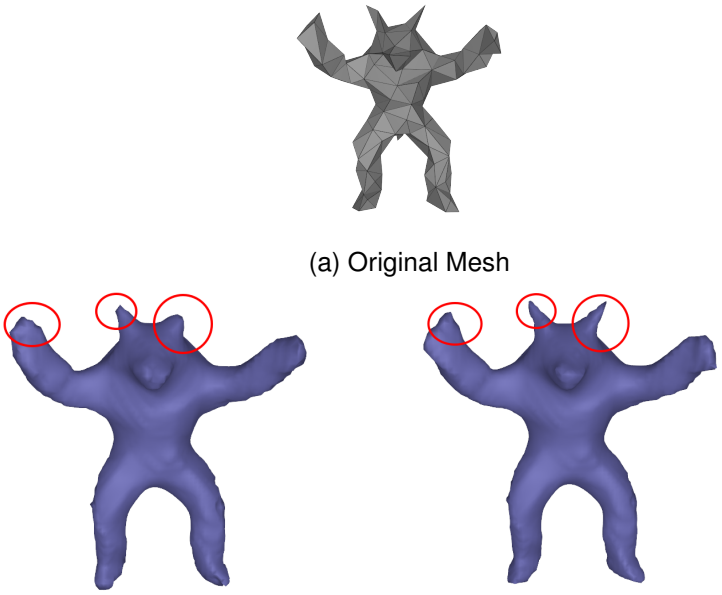


Figure 4.18: The time taken to compute the charge densities across the surface of a mesh deforming under skeletal animation. The motion sequence is a humanoid run cycle lasting 28 frames. Note that the time axis (Y) is a logarithmic scale.

is small. By using the previous frame's charge densities as the initial charge distribution it is possible to reduce the amount of computation significantly and therefore the time. For example, in a 28 frame human run cycle (Fig. 4.17) with an triangulated mesh of 3206 triangles, the time taken for charge computation is 21,381.5ms in the first frame, but by using the previous charge densities as initialisation the mean computation time after the first frame is only 941.9ms (Fig. 4.18).

4.4.4 Accurately Defining the Electric Charge Density: Adaptive Subdivision of the Mesh

In some cases the isosurface at one volt does not accurately represent the mesh surface (see Fig. 4.19, top). This is due to the assumption of uniform charge across each triangle, and generally causes too low potentials in the extrema of the model and areas with sharp edges. A naive solution to this is to iteratively subdivide the model until the size of each triangle is small relative to the overall mesh, thus ensuring the uniform charge assumption for each triangle is nearly true. However this can cause problems with the matrix pseudoinversion charge simulation method when the number of triangles is very large, as it is required to invert a matrix of size $N \times N$ where N is the number of triangles in the mesh (Equation 4.12). Even with the Robin Hood technique additional triangles in the mesh reduces efficiency by unnecessary computation. To alleviate this problem, an adaptive subdivision method is proposed that only subdivides triangles close to the areas where there is a large deviation from the surface potential of one



(a) Original Mesh

(b) Isosurface: 1V (c) Subdiv. Mesh Isosurface: 1V

Figure 4.19: Before (top) and after (bottom) adaptive subdivision is performed on the armadillo man toy. Adaptive subdivision ensures the isosurface at surface potential more accurately represents the true surface of the object. (a) shows the original mesh without subdivision, (b) shows its isosurface at 1V. Note that the extrema are not represented accurately, especially the ears and arms. (c) shows the isosurface of the subdivided mesh at 1V. The isosurface at 1 volt after subdivision represents the object's surface more accurately, especially in extrema (see ears, feet and hands).

volt. The method is given is pseudocode in Algorithm 2. This algorithm iterates until the standard deviation of the potential of samples across the surface is within ε, which was set to 0.05V in these experiments.

The utility of this technique is demonstrated on a non-convex model, the 'Armadillo Man' (Fig. 4.19). The difference between uniform subdivision and the adaptive subdivision method used in this thesis is visible even at low levels of subdivision. With no subdivision the model has 296 triangles. After 6 adaptive subdivision iterations the total number of triangles is 808. With uniform subdivision, the model has 888 triangles after one step of subdivision. By uniformly sampling the potential across the surface, the standard deviation, mean and median of the potential can be computed. As Table 4.1 shows, the adaptively subdivided model more accurately determines the correct potential at the surface with fewer triangles than the uniformly subdivided model.

Input: Object mesh T_{obj}, threshold ε

Output: Subdivided model that allows for accurate representation of the isosurface at 1 volt.

repeat

  Compute the charge distribution over the model;

  forall the $\Delta \in T_{obj}$ **do**

    Randomly sample across the surface of Δ;

    If any sample has a potential deviation greater than ε, mark Δ for subdivision;

  end

  Subdivide the previously marked triangles;

until *No samples have a potential deviation greater than* ε;

Algorithm 2: Adaptive subdivision of a triangulated mesh.

The sum-squared-error is over 6% lower with adaptive subdivision as compared to uniform subdivision, with fewer triangles. This translates to a much improved visual isosurface. An accurate isosurface at the surface potential is essential in order to detect proximity to the surface directly from the potential, and is useful in projection to the polygon surface from the outer space of the object. However, it should be noted that the resulting potential everywhere in the outer space of the object is still harmonic even when the isosurface is not exactly coincident with the true object surface.

Surface Potential	Original Model (296Δ)	Uniformly Subdivided (888Δ)	Adaptively Subdivided (808Δ)
Mean (V)	0.99511	0.99733	0.99789
Median (V)	0.99821	0.99971	0.99992
Standard Deviation (V)	0.01362	0.00926	0.00898
Sum Squared Error (V)	64.00	14.32	13.45

Table 4.1: Adaptive Subdivision Comparison - Analysing the surface potential of 160,000 samples taken across the surface of each model.

4.5 Evaluation of the Electric Parameters

4.5.1 Benchmarking the GPU Implementation of Potential Computation and Flux

In order to lessen the processing time for the electric potential and flux, a parallel implementation for the GPU is presented here, suitable for use with technologies such as NVidia CUDA. The superposition principle allows independent, simultaneous computation of both the potential for each triangle in the object mesh and also for multiple probe points in the outer space of the object. As a result, the computation can be split into $L \times L$ separate equations that can be executed simultaneously via the GPU that each compute the potential between a single probe point and a single triangle. The concept of superposition is then used by taking the sum of the results per triangle to obtain the overall L potentials at the probe points.

To measure the practical difference in the computation time in the GPU implementation as compared to the sequential single-threaded CPU implementation, a regular three-dimensional grid of potentials around the object is calculated for various sizes of grid and the time for completion is recorded. A NVidia Geforce GTX Titan graphics card[7] is used in these experiments, alongside one of the meshes from the database[8] which has 2000 triangles. Results are shown in Table 4.2, and show that as the number of points in the grid increases, the decrease in the runtime by using the GPU implementation over the CPU implementation is approximately 95%.

Computing the electric flux using the GPU gives a similar speedup as shown in Table 4.3. This speedup is realised in a similar way: by splitting the computation into $n \times L$ equations, where n are the number of triangles in the flux measurement surface. Each equation computes the flux between one triangle on the object mesh and one triangle in the flux measurement surface. These can then be summed to produce the overall flux through the measurement surface given the object.

4.5.2 Using the Electric Field In Graphics and Animation

The relationship description given by the Electric Parameters is useful for general design and control of systems relative to objects. To demonstrate that, this section explores two applications of this technique in the fields of graphics and animation: The

[7] This card has 2688 CUDA cores and a base clock rate of 837 Mhz.
[8] Model 164: the teddy in the rest pose is used here.

Num. Samples	CPU Time (ms)	GPU Time (ms)	CPU Time/sample (ms)	GPU Time/sample (ms)
125	456.166	30.8812	3.64	0.247
1000	3525.19	172.729	3.52	0.172
8000	27543.2	1523.80	3.44	0.190
35937	123780	6653.63	3.44	0.185
125000	433901	22809.0	3.47	0.182
1000000	3434800	180882	3.43	0.180

Table 4.2: Timing comparison of the potential computation of a number of sample points around a mesh of 2000 triangles on the GPU as compared to the CPU.

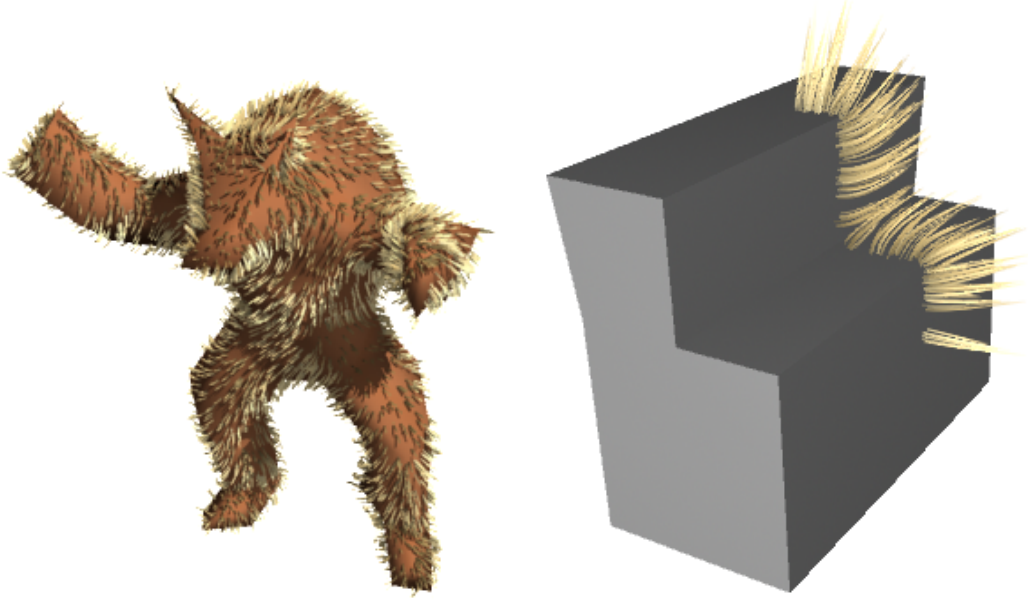
coordinate system can be used for 2.5D control, such as the creation of hair and grass systems without collision, and the potential field can be used for dense correspondence between objects. The electric flux has also been used in the control of cloth [Wang et al., 2013a].

4.5.2.1 2.5D Control

The electric field representation is useful in many graphics applications, due to its ability to define the relative location in the outer space in three-dimensional coordinates to an object's surface. This object-centric coordinate system allows a user to create paths relative to the object for animation quickly and easily. This is briefly described here. For example, particle systems that orbit an object or character can easily be created by ensuring that the potential of each particle stays constant (between 0 and 1) whilst moving in the projected θ and ψ coordinates (termed the UV space).

Systems such as grass, hair and fur can be initialised using this coordinate system also. First, samples can be distributed across the model surface. By decreasing the potential from the surface and keeping the UV values constant, paths that trace the electric field out of the object from the samples can be created. These are guaranteed to be non-intersecting, which makes the initial paths suitable for physical simulation. These can also be controlled in the tangent space to the potential, allowing hair or grass to curl or spiral as it grows. Examples of grass and fur grown using this technique is shown in Fig. 4.20. The grass successfully avoids the virtually charged objects in the scene. The fur on the model is sparsely shown in Fig. 4.20b in order to enable the curl

(a) Grass avoiding obstacles.



(b) Fur samples grown using the electric field.

Figure 4.20: An example of the coordinate system of the Electric Field used (a) with multiple objects to grow grass, and (b) to avoid self-penetrations whilst maintaining artistic control during fur creation.

Num. Tri in T_{obj}	CPU Time (ms)	GPU Time (ms)	CPU Time/tri (ms)	GPU Time/tri (ms)
20	0.28	1.58	0.0141	0.0792
536	7.03	2.33	0.0131	0.00436
800	10.47	2.58	0.0130	0.00322
1024	13.09	2.49	0.0127	0.00243
2000	25.70	3.39	0.0128	0.00169
4356	57.29	4.29	0.0131	0.000985
7964	103.61	5.19	0.0130	0.000652

Table 4.3: Timing comparison of the flux computation on the GPU as opposed to the CPU-only version. We use a mesh of 20 triangles for the flux measurement surface and vary the number of triangles in the charged model. The GPU implementation is around 20× faster than the CPU when the charged model has around 8,000 triangles.

of the individual hairs, caused by a tangential term, to be seen.

4.5.2.2 Filling Holes for Object Correspondence between Objects of Arbitrary Topology

There are additional uses to the electric field aside from describing the spatial relationship. Topological changes occur in the isosurface as the potential varies, which allows versions of the shape with different genus to be created. The isosurfaces have dense correspondence to the original object surface, allowing the isosurface to be used as a proxy mesh for certain correspondence techniques. As topological complexity of the model often causes current global correspondence techniques to fail [Kim et al., 2011; Kaick et al., 2011], this proxy mapping can be useful for dense correspondence generation between arbitrary objects.

All objects have an isosurface in their electric potential field that is genus zero: as the potential decreases the topological holes eventually disappear. Decreasing the potential of the isosurface fills in holes quickly as compared to deformation of the shape further from the holes as the potential varies quickly inside topological tunnels. The isosurface at which point the shape topology changes whilst the shape remains similar to the original object can be found using binary search on the potential. This provides a version of the shape which closely approximates the original object but is

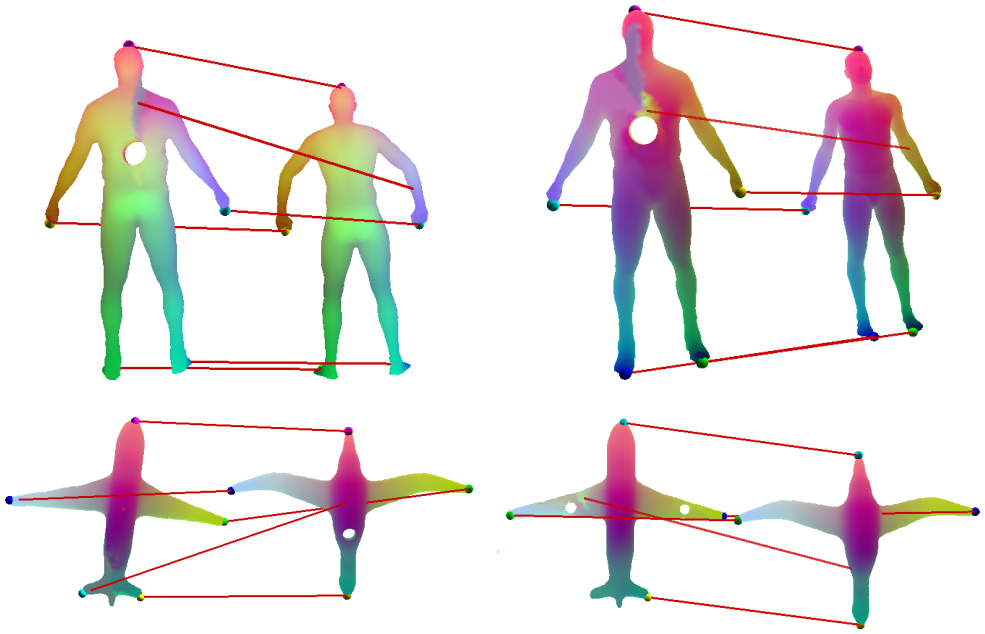


Figure 4.21: Blended Intrinsic Mapping errors: here the mapping is shown via colour transfer from the right object in each example to the left object, generated by their code using their original colour scheme. Errors occur near and around mapped holes when Blended Intrinsic Mapping is used directly with objects that are not genus zero.

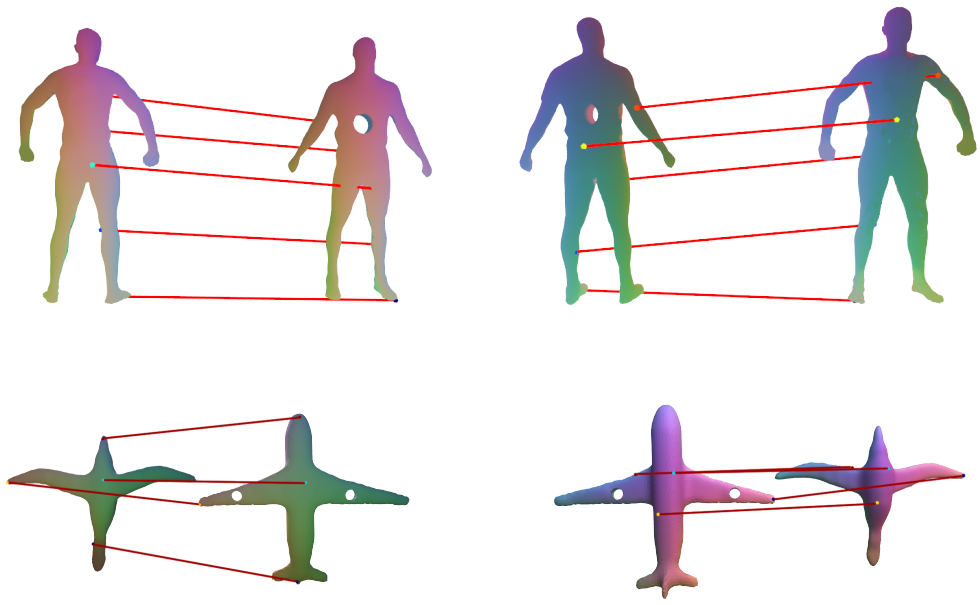


Figure 4.22: Automatic correspondence using electric isosurfaces as proxy meshes.

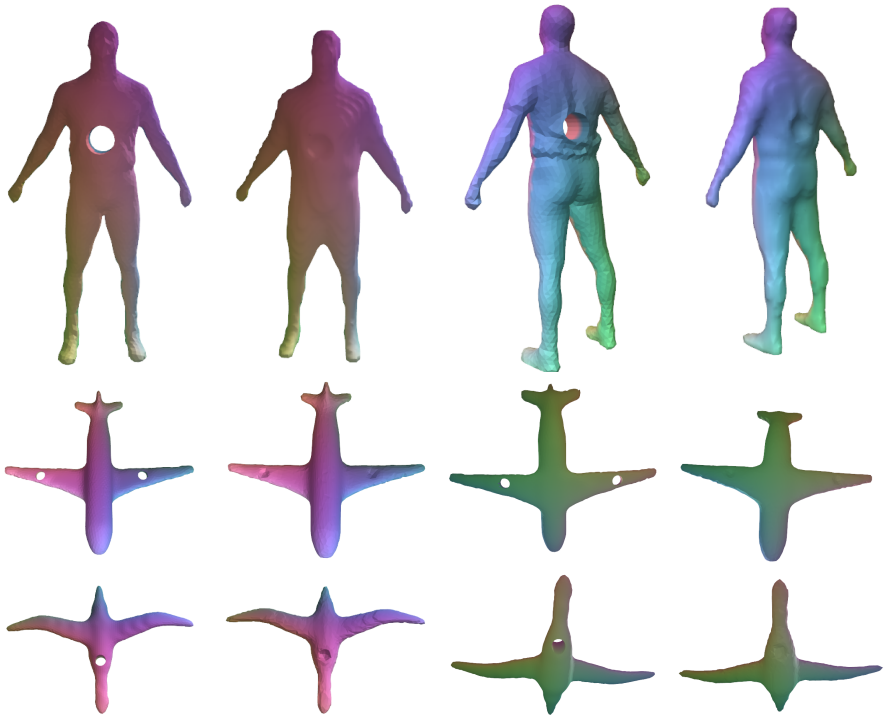


Figure 4.23: Models with holes and their corresponding closed surfaces.

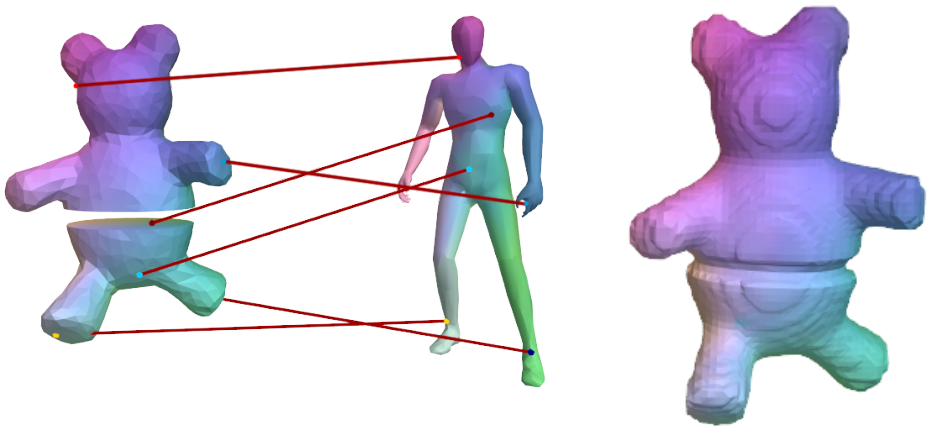


Figure 4.24: Automatic correspondence of a split model (left) to a connected model (centre) using the genus zero isosurface (right).

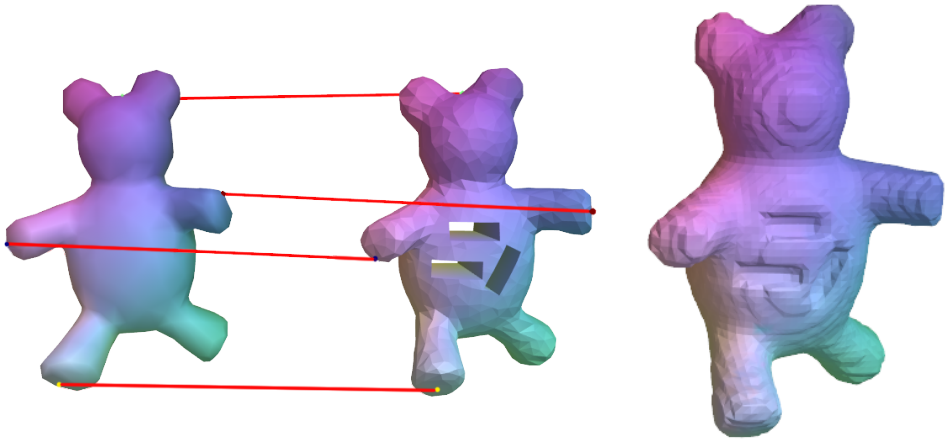


Figure 4.25: Automatic correspondence of a model without holes (left) to a similar model with multiple large holes (centre) using the genus zero isosurface (right). Note the lack of error around the holes.

genus zero.

Filling in holes is useful as it is easy to make topologically complex models either manually or by reconstruction from scan data due to occlusions, and has been studied extensively. Other techniques have been proposed to automatically fill in holes in models: voxelisation and Radial-Basis Functions (RBFs) have been used successfully in repair of models acquired from range data[Davis et al., 2002; Carr et al., 2001]. However, these replace the original geometry of the model, with the assumption that the newly created model is more accurate. This assumption is not necessarily the case when the object truly is topologically complex (as it may be if created by an artist). Furthermore, in these cases of hole filling there is no clear correspondence between the original mesh and the mesh with its holes filled. However, although the technique presented here also fills in topological holes, it has a separate objective to most hole repairing work as this thesis considers object correspondence: specifically the production of a dense correspondence between arbitrary objects by filling in both holes and tunnels in the mesh, as well as connecting disconnected components, to create a genus zero proxy object. Practically there are a variety of applications for dense correspondence between objects of arbitrary genus that have a limited number of seams including (but not limited to): transfer of textures, morphing of shape, and creation of a parametric shape space between different models. This thesis primarily uses the correspondence to transfer the electric coordinates between different objects, thus preserving the spatial relationship.

Input: Object mesh T_{obj}, per-triangle charge densities Q_{obj}, surface parametrisation S (generated from the Electric Coordinates), ε target precision

Output: Genus zero mesh T'_{obj}, corresponding surface parametrisation S'.

```
// Upper bound
```
$I_u = 1;$
```
// Lower bound
```
$I_l = 0;$

while $abs\ (I_u - I_l) > \varepsilon$ **do**

  $I_{mid} = \frac{I_u + I_l}{2};$

  `// Computes the isosurface at` I_{mid} `given the original mesh and charges`

  T'_{obj} = Isosurface(I_{mid}, T_{obj}, Q_{obj});

  `// Computes the first three Betti numbers for a mesh.`

  β = Betti(T'_{obj});

  if $\beta == (1,0,1)$ **then**

    `// Homeomorphic to a sphere`

    $I_l = I_{mid};$

  else

    `// Not homeomorphic to a sphere`

    $I_u = I_{mid};$

  end

end

T'_{obj} = Isosurface(I_l, T_{obj}, Q_{obj});

`// Computes the charges for a mesh. This could be (for example) the Robin Hood method as described in Algorithm 1.`

Q' = ComputeCharges(T'_{obj});

`// Computes the Electric Coordinates at the surface of the mesh, as described in Section 4.4.3 and in further detail in Section 5.3.1`

S' = SphericalMap(T'_{obj},Q');

Algorithm 3: Hole-filling with correspondence to the original mesh.

Approach In order to compute the dense correspondence between objects, a genus zero representation of the object must be found with correspondence to the original object. This is outlined in Algorithm 3. In practice, an isosurface of the electric field can be taken close to the object as both an approximate representation of the original shape of the object and also closed with genus zero (has no topological tunnels or disconnected components). However the potential of this isosurface varies between object models. Finding the isosurface with greatest potential that is genus zero gives the closest representation to the original object without topological holes. In order to do this, a valid assumption is that the function is smooth and so binary search can be used. As the surface potential is 1 and the potential drops to 0 at infinite distance, the bounds of the potential of the isosurface are $(0, 1]$, and a boundary exists in the range that separates genus zero isosurfaces from isosurfaces with topological holes. Finding this boundary is possible by computing the genus of the isosurface for the midpoint of the lower and upper limits using the Betti numbers [Barile and Weisstein, 2014], where specifically β_1 must be zero, and β_0 and β_2 must be one for the isosurface to be homomorphic to a sphere[9]. The search range is then reduced to the greater half if the genus is zero, otherwise it is reduced to the lesser half. This process is iterated until the required precision is achieved, and the final genus zero isosurface closely representing the original shape can be produced.

This isosurface has a correspondence to the original object at any point on its surface by following the harmonic field in the positive gradient towards the original object surface, but equivalently can be computed by tracing points from the isosurface in the negative gradient (away from the object), and taking the surface parametrisation for the isosurface (as the object surface was above) by the endpoints of these traced lines. Either of these techniques gives a correspondence between the electric coordinates of the object surfaces.

To compute a correspondence between objects, the genus zero isosurface is used as a proxy object to compute the Blended Intrinsic Map (BIM) [Kim et al., 2011] between the original object and the target. As the mapping between the proxy, genus zero object and the original object is known, the BIM can be computed between the proxy and the target, giving the correspondence between the original object and the target.

[9]Informally, the Betti numbers are topological invariants: β_0 describes the number of connected components, β_1 describes the number of one-dimensional holes (cuts), and β_2 describes the number of cavities (enclosed regions) in the mesh. Marching Cubes [Lorensen and Cline, 1987] is used to create the triangulation of the mesh from which the Betti numbers can be computed [Delfinado and Edelsbrunner, 1993].

Experiments In order to validate this technique, a set of five models with holes and topological irregularities are created using Constructive Solid Geometry (CSG) techniques alongside the TOSCA [Bronstein et al., 2008] and Watertight [Giorgi et al., 2007] 3D model datasets (Fig. 4.23). These models are unable to be processed by BIM in the case of disconnected components, or cause major errors near to the holes (Fig. 4.21). This is a known limitation which is pointed out in the original publication. These experiments look to map these difficult examples to novel objects and evaluate the semantic quality by the similarity to the true mapping on genus zero objects given by BIM.

By computing the electric field, the two-dimensional parametrisation of the surface known as the UV coordinates is created, which can be used for texture mapping and to define the surface correspondence. The genus zero shape representation is created from this field, with inherent correspondence to the original model given by the field lines connecting the two.

A selection of the models and their correspondences are shown in Fig. 4.22. These do not have the obvious visual correspondence errors as if the topologically complex models were used directly and still have correct correspondence to the rest of the model.

In Fig. 4.24, the teddy object has been split across the centre. A correspondence to another object is not possible to be automatically computed using Heat Kernel [Ovsjanikov et al., 2010], Möbius Voting [Lipman and Funkhouser, 2009] or other similar techniques directly. However, by creating the genus zero isosurface of the split object, the isosurface can be used as a proxy for these techniques. The results are shown in this images as corresponding colour, with the correspondence between the isosurface and original model given by the electric field, and the correspondence between the isosurface and the target model given by the technique of Kim et al. [2011]. Similarly, it is shown that this technique works well even when there are multiple, larger holes in the mesh (Fig. 4.25).

To quantitatively evaluate the correspondence, a comparison between the Blended Intrinsic Map on the original, genus zero model prior to the CSG operations (as the ground truth) and the mapping proposed here for the object with topological holes is computed. There is a natural mapping from the genus zero model and the modified CSG model by vertex correspondence (all vertices will directly correspond except in areas with holes, which do not have correspondence). Evaluating the difference in the mapping between these two and the target model gives the deviation of the new

	$E(c_1,c_2)/10^{-2}$					
bird (hole)	2.95	3.55	4.25	3.99	4.18	4.25
plane (holes)	3.43	2.31	4.21	3.81	3.15	3.42
teddy-1 (holes)	4.25	3.50	3.95	2.57	3.01	3.89
teddy-1 (split)	3.81	3.17	2.81	3.42	2.90	3.08
man-1 (hole)	3.47	3.86	3.32	3.97	3.48	3.50
$\mathbb{M}_1 / \mathbb{M}_2$	bird	plane	teddy-2	teddy-3	man-2	man-3

Table 4.4: Deviation between the ground truth and our mapping. Note that these values are not similarity values, rather deviation from Blended Intrinsic Mapping when using the electric field isosurface.

mapping from the original BIM. As in the evaluation by Kim et al. [2011], the measure of similarity is computed as the normalised geodesic distance across the mapped object surface for each vertex with direct correspondence in the genus zero and non-genus zero models, given by:

$$E(c_1,c_2) = \frac{\sum_{v\in\mathbb{M}_1} g(c_1(v),c_2(v))}{\sqrt{area(\mathbb{M}_2)}} \tag{4.13}$$

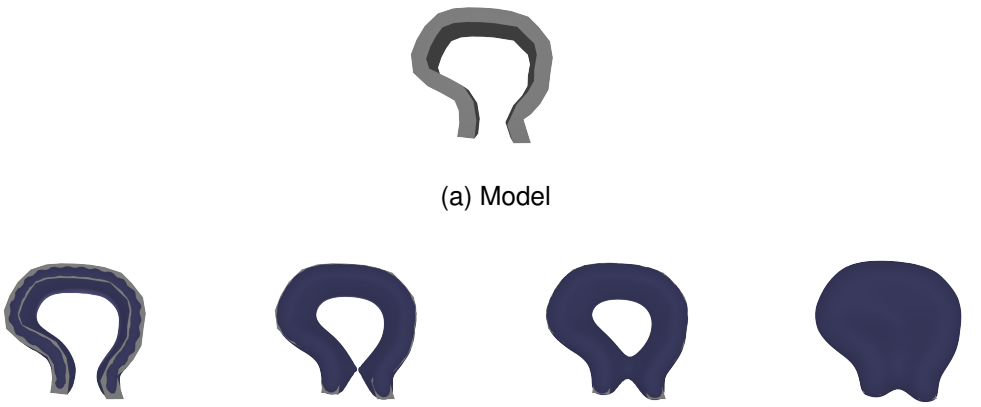
where $\mathbb{M}_1$ is the initial object and $\mathbb{M}_2$ is the target object, $E(c_1,c_2)$ is the error between mapping $c_1 : \mathbb{M}_1 \to \mathbb{M}_2$ and mapping $c_2 : \mathbb{M}_1 \to \mathbb{M}_2$, and the function $g(a,b)$ is the geodesic distance between points a and b on $\mathbb{M}_2$.

Numerical results of the comparison between the technique outlined in this chapter and BIM on the original, genus zero model are shown in Table 4.4. They show that the mapping using the new technique varies only slightly from the true original mapping (which here is considered as the Blended Intrinsic Map) whilst enabling mapping of objects of arbitrary topology. The variation is likely due to the smooth stretching of the correspondence near holes. In comparison, BIM had a E value of around 0.05 from the ground truth mapping in their experiments in the best cases [Kim et al., 2011].

4.5.2.3 Finding Singular Points in the Field

The singularities of the electric field are located in areas of the mesh that either contain topological tunnels, are in-between multiple disconnected charged objects in the scene, or in cases where there are severe concavities. By identifying the location of these singularities, these locations can be used, for example, in path planning problems such as winding the fingers of a hand around handles, or in connecting objects

using implicit surfaces similar to metaballs [Blinn, 1982] by determining the potential at these stationary points, giving the minimum isovalue required for connection.



(a) Model

(b) 1 volt (c) 0.96 volts (d) 0.95 volts (e) 0.91 volts

Figure 4.26: Singular points example: As the isosurface decreases from 1 volt (b) to 0.95 volts (d), the topology changes. Note that this is a three-dimensional example, and there is no local minimal point but rather a stationary point in the centre of the object. Eventually the object becomes genus zero again at 0.91 volts (e).

Singularities in the electric field Singularities in the field are where the magnitude of the field is zero, and so the instantaneous direction of the field is undefined. As the field is harmonic, there are no local minima or maxima in the field, or the field would be a constant value everywhere [Berenstein and Gay, 1997]. Stationary points can occur however, and most notably do when the generating mesh has topological tunnels or multiple disconnected components. However, stationary points can also appear when a genus zero mesh is heavily concave, as shown in Fig. 4.26. This figure shows that as the potential decreases, the isosurface changes topology. At the point where the genus zero isosurface becomes genus one (the two surfaces 'touch' creating a topological tunnel), there is a singular point that has partial derivatives of zero. This can be seen when considering the field at the isosurface. By definition, the gradient of the field must be in the direction of the expansion of the isosurface (the normal direction on the surface). As the two surfaces approach, this vector becomes zero as the distance approaches zero.

In practice, these points do not directly affect the projection to the surface of the object, as they are individual points and have a Lebesgue measure of 0 in the set of

points in the external space[10]. Accordingly a field line for a randomly chosen point in space will not trace its electric field line through these points. However they are a useful descriptor for the topology of the electric field, and the aim is to make use of them by detecting their location.

Locating the singularities In order to find the approximate location of these singularities, the outer space of the object can be sampled in a similar manner to the detection of 'pits' (local minima), 'passes' (stationary points) and 'peaks' (local maxima) in geographical data by Takahashi et al. [1995], extended to 3 dimensions. Takahashi et al.'s technique uses a sampled regular two-dimensional heightmap and proceeds as follows. For any sample, the 8 neighbouring points can be split into two groups: $+ve$ that are points higher or at the same height than the sample point, and $-ve$ that are points lower in height than the sample point. The boundaries between these classes of neighbouring points can be computed. If there are two boundaries, the point is a 'normal' point, not a singular point (the field flows in some direction through this point). If there are four boundaries, then the sampled point must be singular. In the case of no boundaries (i.e. either all neighbouring points are greater or all neighbouring points are less than the sample point) then the sampled point is a local extrema. By classifying the samples in this way, the singularities can be found by solely checking neighbouring potentials. Algorithm 4 shows this in pseudocode form.

The insight in extending this to support a three-dimensional real-valued function is that there must be only three disconnected regions neighbouring a singular point to determine it as a stationary point, as the additional dimension allows for further connectivity between regions. In this case, the 26 neighbouring points (Fig. 4.27) can be checked and clustered into the two groups based on the grid connectivity. By checking the number of neighbour clusters, we can define each point as either a stationary point (has 3 discrete neighbour clusters), a local extrema (has 1 neighbour cluster of either lower or higher value than the considered value), or neither (any other result). Local extrema do not exist in the electric field, so we are left with discriminating between stationary points or not. Examples of detected stationary points are shown in Fig. 4.28.

These singularity locations are not currently utilised in the manipulation transfer system discussed in Chapter 5, but could be useful for generation of novel grasps that pass through handles, without required input trajectories.

[10]This can be thought of as the probability of selecting a random point in the outer space of the object and it being a singular point is zero.

Input: Object mesh T_{obj}, per-triangle charge densities Q_{obj}
Output: The locations **L** and classes **C** of each singular point
Initialise **L**, **C**;
forall the *points p in grid surrounding the object* **do**
  n = Neighbours of p;
  if $\forall n, V(n, T_{obj}, Q_{obj}) < V(p, T_{obj}, Q_{obj})$ **then**
    C.push(p is a local maxima);
    L.push(Location(p));
    continue to next p;
  end
  if $\forall n, V(n, T_{obj}, Q_{obj}) > V(p, T_{obj}, Q_{obj})$ **then**
    C.push(p is a local minima);
    L.push(Location(p));
    continue to next p;
  end
  $n_+ = [n : V(n, T_{\text{obj}}, Q_{\text{obj}}) > V(p, T_{\text{obj}}, Q_{\text{obj}})]$;
  $n_- = [n : V(n, T_{\text{obj}}, Q_{\text{obj}}) \leq V(p, T_{\text{obj}}, Q_{\text{obj}})]$;
  forall the *points* $x \in n_+$ **do**
    forall the *neighbours* x' *of* $x \in n_+$ **do**
      Add the neighbours of x' to the neighbour list of x;
      Delete x' from n_+;
    end
  end
  forall the *points* $x \in n_-$ **do**
    forall the *neighbours* x' *of* $x \in n_-$ **do**
      Add the neighbours of x' to the neighbour list of x;
      Delete x' from n_-;
    end
  end
  if *Count* n_+ + *Count* n_- == *3* **then**
    C.push(p is a stationary point);
    L.push(Location(p));
  end
end

Algorithm 4: Singular point detection algorithm.

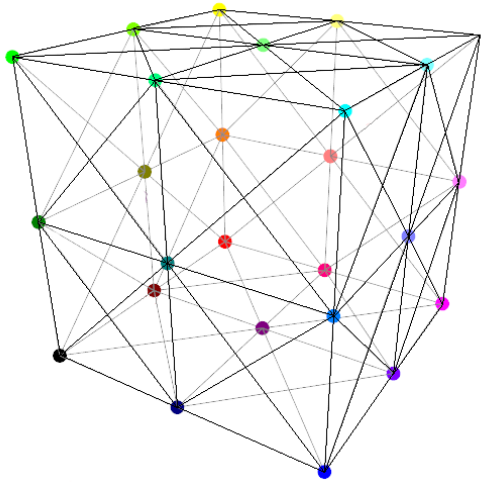


Figure 4.27: The connectivity of the 26 neighbours around the sample point (not shown).

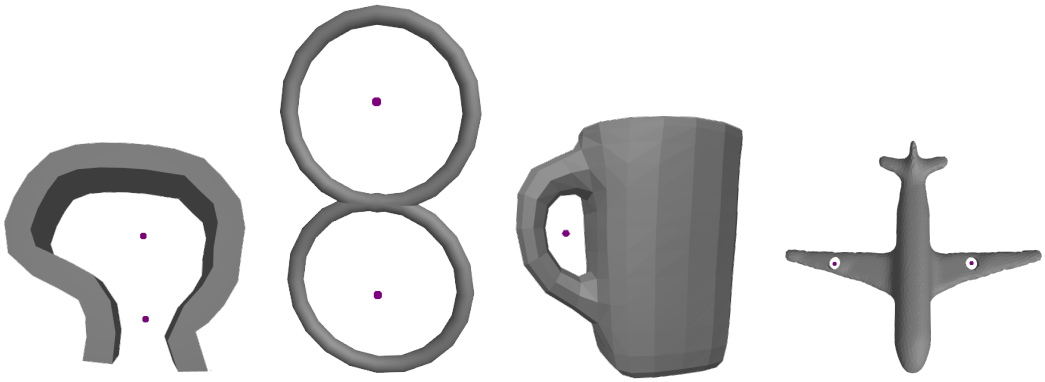


Figure 4.28: Examples of detected stationary points for the electric field, shown in purple.

4.6 Chapter Discussion

In this chapter the merits of a representation that can encode both relative transformation of points and envelopment of a target by a surface have been presented; an extension to the Interaction Mesh to deal with objects and an envelopment metric for a point in three-dimensions has been shown; and the proposed system of Electric Parameters has been described and evaluated in terms of its description of spatial relationship.

The envelopment of a point by the winding number in three dimensions (as described in Section 4.2) is computationally efficient, but limited in terms of the problem of grasping, only being valid for simpler (convex) objects.

The extension of the Interaction Mesh to objects does not take into account the

surface of the object directly and so does not allow for adaptation of the motion (such as novel contacts or new contact locations) when the motion is applied to a new object. Moreover, the use of the Euclidean distance metric between target poses is problematic when the previous location of the hand cannot be directly interpolated to the new location. Further collision detection and smoothness terms can be added to alleviate these problems, along with additional samples in the space, but this then leads to a complex cost function.

The Electric Parameters defined in this chapter are computationally more complex, but define both the envelopment for arbitrary objects and the spatial relationship in a coordinate system that naturally avoids collision with the object and is defined for the entire outer space of the object. Projection of a point in the outer space to the object surface is defined by the UV coordinates, and a measure of distance from the object along the field line is given by the electric potential. This representation is shown to be useful in the field of computer graphics by examples of 2.5D control and by dense correspondence generation, as the electric field alongside correspondence techniques such as Blended Intrinsic Mapping can be used to intuitively map between objects of arbitrary topology, meaning that transfer of the spatial relationship to novel objects is possible. This is utilised in the next chapter, which considers generation and transfer of grasps by robotic manipulators.

Chapter 5

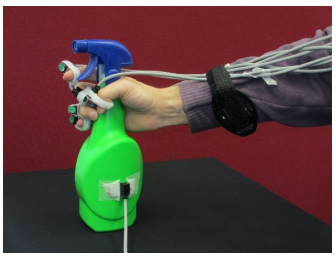
Generation and Reproduction of Interaction: Controlling Robots for Transfer of Interaction using the Electric Field

5.1 Introduction

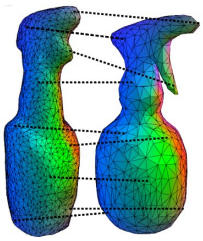
This chapter discusses the generation of grasping motion given the interaction capture technique presented in Chapter 3 and the representation of interaction described in Chapter 4. Some of the results in this chapter have appeared in print previously [Sandilands et al., 2013b].

Some of the most complex motions of interaction a human can perform are manipulations of objects. Movements that involve reaching and grasping are difficult to synthesize, due to collisions between the hand model and its environment, self-collisions, and the complex shape of the open space, especially when concave objects are involved. Synthesis of such movements is expensive because collision detection and global path-planning are required. Furthermore, the movement is no longer valid once the geometry of the hand or the shape of the object to be grasped changes, or if the situation changes.

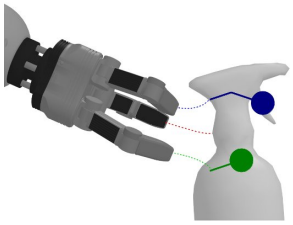
A similar problem is evident in applications such as teleoperation [Hu et al., 2005] or learning from demonstration [Kang and Ikeuchi, 1994], where the aim is to compute the mapping between the human hand and the robotic hand that have different geometrical or even topological structures. This mapping is usually non-linear and



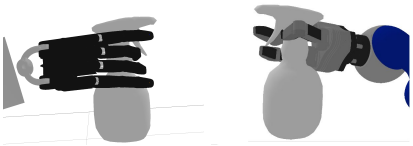
(a) Motion capture.



(b) Object correspondence.



(c) Electric parameter preservation.



(d) Grasp transfer to new object and/or hand.

Figure 5.1: Synthesizing a grasping motion from motion capture. (a) An interaction is captured of an actor grasping the object. (b) A dense mapping to a new object is produced using the electric field. (c) The interaction points on the hand projected to the object's surface by following the electric field gradient. The solid line is the path on the surface that the projection must match for the rest of the sequence. (d) The re-targeted motion.

is computed manually by exploiting the task definition using a proxy object [Gioioso et al., 2013], by finger tip position correspondence [Peer et al., 2008] or by using machine learning techniques with example pose correspondences [Ciocarlie et al., 2007]. These ways of mapping the motion either restrict the mapping in terms of complexity of the motion or dissimilarity of the kinematics of the two systems (in the former case), or they require a large amount of sample data (in the latter case).

In this thesis these problems are addressed by using the novel representation of Electric Parameters (defined in the previous chapter) to specify the spatial relationship between the hand and the object, including the envelopment of the hand around the object. Firstly, by virtually charging the object, the object-centric coordinate system termed the *electric coordinates* is computed, as described in Section 4.4.3. The electric coordinates can canonically parametrise the open space around objects. The movements of the hand model defined in the electric coordinates can be parametrised even when they involve extensive close interaction between the hand model and the

object. This is because the relationship is defined between the hand and object geometry itself rather than samples in the space. Also, due to the harmonic nature of the electric coordinates, the hand model can be easily guided towards the target location on the object without suffering from local minima. Secondly, the electric flux is used along with Gauss' law to quantify how much the object is surrounded by the hand model. This allows for abstraction of the envelopment of the hand around the object, and successful grasping by use of the electric flux as a control parameter.

Using this representation, the complexity of motion planning and transfer can be significantly reduced by offloading the computational effort onto the mapping between spaces, which can be easily achieved using the mapping techniques introduced in Section 4.5.2.2. The complexity of the space for motion planning is lowered by abstracting the movements from those in the joint angle space to those in the electrostatic parameters. This mapping enables the user to apply optimisation methods with *local* motion planning, where exploratory methods such as RRT[1] or PRM[2] would be necessary otherwise. The optimal trajectory is computed within the Approximate Inference Control framework (AICO)[Toussaint, 2009] that allows the combining of the electrostatic representation with other representations such as the control effort in joint space. Together these contributions create a technique that transfers not only the grasp but the natural approach to the object between different manipulators that is both physically realistic and preserves the intent of the original motion. This preservation of intent is useful for programming by demonstration with few examples required, and in animatronics to create artistic motion relative to objects or to preserve a style of motion. Experimental results provide validation that the proposed representations are simple enough that local optimisation methods such as AICO can be used. Further experiments demonstrate direct grasp transfer/teleoperation with different biases and styles.

Contributions:

- A novel technique for transfer of motion relative to an object, suitable for teleoperation and programming by demonstration, using local optimisation alongside the Electric Parameters.

- Experiments showing grasp transfer with different biases, multiple manipulators, and objects from the interaction database.

[1] Rapidly-exploring Random Trees[LaValle, 1998]
[2] Probabilistic Roadmap Method[Kavraki et al., 1996]

5.2 Overview

This chapter will cover topics including grasp synthesis, grasp transfer, and potential-field guided path planning[3]. The work explored in this chapter has simularities to potential-field guided methods, as the electric field and potential are created using charge simulation, which is then applied for guiding the hand model around the object. However, in contrast to other potential-field methods, this work makes use of the field as a coordinate system, not to follow the gradient to reach a goal but for generalisation of the grasping behaviour, which is useful for transferring the grasping motion to novel situations (such as in teleoperation situations).

This chapter considers the task of interaction transfer as the problem of transferring the relative configuration of the manipulator and the object. Grasping and dexterous manipulation[4] is important in close interaction and so this chapter focuses on this aspect of the task. The combined hand and arm model is defined as a kinematic system with N number of controllable joints in configuration q where the positions of M number of points $\vec{p}_i(q)$ attached to a link connected to the system by joint j is defined by the forward kinematics function $f_j(q)$:

$$\vec{p}_i(q) = f_j(q), q \in \mathbb{R}^N. \tag{5.1}$$

The points $\vec{p}_i(q)$ are used to both approximate the desired contact surface of the hand (via W number of triangles, see Fig. 5.4), and to represent the positions of joints in the hand (see Fig. 5.1c). Triangulation is used to approximate the shape of the object using K number of points $\vec{p}_k$ and L number of triangles $\Delta \in T_{\text{obj}}$(see Section 4.4.3.1 for related details of the mesh representation). The goal is to generalise the relationship between points $\vec{p}_i(q)$ and $\vec{p}_k$ so that the joint configurations q produce consistent, stable and semantically similar reaching and grasping motion when the geometry of the hand model changes (changing the forward kinematics function $f_j(q)$) or when the shape of the object changes (changing $\vec{p}_k$ and T_{obj}).

The hand-object interaction is generalised using properties of a virtual electrostatic field defined using the vertices $\vec{p}_k$ and triangles T_{obj} of the object mesh. Once the motion is defined in the abstract representation, it can be easily transferred to different hand models and objects. The semantic similarity is encoded within the representation through joint position in the electrostatic coordinates and through hand envelopment of the object using the electric flux. The trajectory for new hand model is computed using

[3]Related work for these areas is explored in Section 2.3.3

[4]The ability to translate and rotate the object arbitrarily.

the stochastic optimization framework AICO, such that the hand model follows the desired trajectories of the electrostatic parameters as much as possible, while avoiding collisions with other objects in the scene, minimising the control effort, and satisfying physical constraints.

Section 5.3 describes how the relative state of the hand is encoded using the Electric Parameters, and the technique for computing force closure for projected points onto the object surface is given. The difference in these states from the target states is used as an objective function. Specific implementation details are given in Section 5.5. Section 5.4 outlines the cost function that we use in combination with AICO to produce the robot trajectory. Finally, the results of a variety of motion transfer experiments between the human hand and various robots, as well as between different objects, are shown in Section 5.6.

5.3 Hand Representation using the Electric Parameters

This section describes the representation of the interaction using the Electric Parameters. The parameters are useful for defining the reaching and grasping motion with respect to the object as it gives a relative spatial relationship between points or surfaces and a charged object. This section assumes that the Electric Parameters for the object have been computed as described in Chapter 4, giving L charges $\mathbf{Q}_{\text{obj}}$, where L are the number of triangles belonging to the object, and $\mathbf{Q}_{\text{obj}}$ are the charge densities per triangle.

In Section 5.3.1 the description of the relative configuration of the hand is defined in the electric coordinates. In Section 5.3.2, the cost function that computes the grasp quality for the expected grasp is discussed. Finally in Section 5.3.3, the hand envelopment is encoded using electric flux.

5.3.1 The Relative Configuration of the Hand

In order to preserve the spatial relationship between the hand and the object, it is necessary to establish the relative configuration of the hand, which is done by the electric coordinates. The system developed for this process is agnostic to human hands or robotic manipulators: as long as the points $\vec{p}_i(q)$ can be defined, the interaction can be encoded.

The relative configuration of the hand is recorded using the electric coordinates for

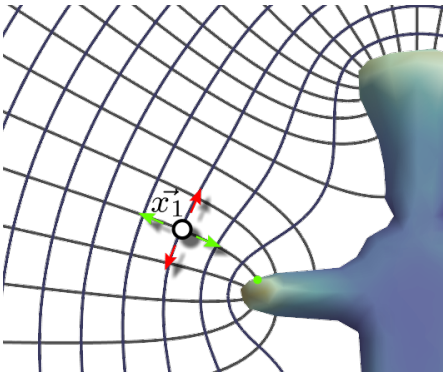


Figure 5.2: A two-dimensional example of control in the electric coordinates for $\vec{x_1}$. Varying the target UV moves $\vec{x_1}$ in the direction of the red arrows, whilst varying the target potential moves $\vec{x_1}$ in the direction of the green arrows.

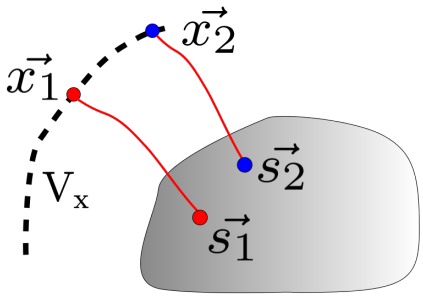


Figure 5.3: An illustration of the projection of points $\vec{x_i}$ onto the surface of an object directly using the electric coordinates. V_x is the isosurface (surface at equal potential, dashed line) of the electric field that $\vec{x_i}$ lies on, $\vec{s_i}$ is the point on the surface of the object corresponding to $\vec{x_i}$ obtained my increasing the potential along the electric field line that intersects $\vec{x_i}$ (shown as a solid red line).

each selected joint, at each frame of the animation. The orientation of the selected joints is also encoded in this manner, by attaching points at small offsets in the local X, Y, and Z components of each joint's coordinate system. In order to increase the user-controllability of the system, the coordinate state over the hand joints are split into two components: ϕ_{uv}, which are the two-dimensional projected point on the surface of the object, and ϕ_p, the potential of the point relative to the object (giving a measure of control of the distance from the surface, as seen in Fig. 5.2).

ϕ_p is computed simply as the potential of each of the points attached to the hand:

$$\phi_p(q) = \begin{bmatrix} V(\vec{p}_0(q), T_{\text{obj}}, Q_{\text{obj}}) \\ V(\vec{p}_1(q), T_{\text{obj}}, Q_{\text{obj}}) \\ \vdots \\ V(\vec{p}_M(q), T_{\text{obj}}, Q_{\text{obj}}) \end{bmatrix} \tag{5.2}$$

The function V that computes the potential of a single point given an object and its charges is defined in Section 4.4.3.1. This gives the unique location along each electric field line generated from the object that the points $\vec{p}_i(q)$ lie on.

To find the corresponding point on the object's surface given a point in the outer space of the object, the field line (gradient of the potential) of the electric field can be traced. The object surface has a two-dimensional parametrisation given by the electric field called the UV coordinates, as discussed in Section 4.4.2. This is the normalized elevation and azimuth values of the end points of the field line that originates from each point on the object surface, relative to the object transformation. In practice the projected sphere that created the object surface UV coordinates is not truly infinite, as it is impossible to trace the field lines in that way. Instead, to compute these coordinates in the outer space the electric field lines are traced to a *low* potential (0.5V in our experiments)) giving endpoints $\vec{x}$, which ensures the endpoints are far from the object so that the local object geometry does not affect the sphere significantly. The UV coordinates at the object surface are pre-computed from $\vec{x}$ as $(\arccos\frac{z}{r}, \arctan\frac{y}{x})$, where x, y, z are the Cartesian coordinates of the projection of $\vec{x}$ by the electric field to the sphere, and r is the radius of the sphere, which is the distance from the object's centroid.

ϕ_{uv}, the projection onto the surface of the object for each point attached to the hand $\vec{p}_i(q)$, may then be computed by integration along the field line toward the surface. ϕ_{uv} is computed numerically by evaluating the path integral of the electrostatic field along the field line C that passes through a point $\vec{x}$ via Runge-Kutta (RK4) integration (see

Fig. 5.3).

$$f_{uv}(\vec{x}) = \int_{V(\vec{x})}^{1} \vec{E}(\vec{x}) d\vec{x}, \text{where } \vec{x} \in C. \tag{5.3}$$

$$\phi_{uv}(q) = \begin{bmatrix} f_{uv}(\vec{p}_0(q)) \\ f_{uv}(\vec{p}_1(q)) \\ \vdots \\ f_{uv}(\vec{p}_M(q)) \end{bmatrix} \tag{5.4}$$

The projected azimuth u and elevation v (from the endpoints of projection $f_{uv}(\vec{x})$) relative to the object are used as the first two parameters in our parametrisation as ϕ_{uv}, and the electric potential ϕ_p is used as a measure of distance along the field line and the third parameter. This provides a projection onto the surface of the object for any point in the outer space of the object, defined by the UV coordinates. The correspondence between different objects is then defined as in Section 4.5.2.2 through correspondence between isosurfaces of the field. This is a bijective mapping defined for all points on the surface of the object where there are no saddle points[5]. A mapping exists even for objects of dissimilar shapes and arbitrary topology[6] due to the non-intersection property of the field lines, but there are seams that are not bijective when there are topological tunnels.

Computing the difference between UV states Computing the distance between UV states ϕ_{uv} can be more complicated that simple spherical distance when the objects have topological holes. In fact, in order for the distance to correctly take into account the tunnels of the object, the topology of the UV space becomes the object's topology itself. Because of this, when dealing with objects that have holes the UV coordinates are used to define the projection to the surface, and then the geodesic distance over the object's surface between the current and target UV can be used as the UV space distance metric. Although this is necessary to reproduce motions that pass through topological holes, many objects are genus zero (and some that aren't are treated as such), and in

[5]When there are saddle points, for example due to the electric field generated by objects with holes such as a torus, these points cause seams to exist in the mapping. Consider the torus example: around the central point of the hole the field diverges and so maps to a seam on the object. These saddle points, however, do not form a volume due to the maximum principle[Berenstein and Gay, 1997]. This states that a volume in the exterior of the object that has zero field would have constant potential, and if this occurs the potential in the whole domain must also be constant. Therefore, either the entire field is constant, or volumes of saddle points (regions of zero field) do not exist in the electric field.

[6]Without self intersections.

(a) The flux surface

Figure 5.4: The flux measurement surface as specified on the human hand model and the Schunk robotic hand model. The envelopment of the object by these hands is reported using this surface.

those cases can be instead approximated by a sphere for efficiency reasons. This allows the UV space distance metric to be simply the spherical distance in these cases.

5.3.2 Force Closure

The projection onto the surface of the object also makes it possible to compute force closure directly from surface points projected using electric coordinates

$$\phi_{fc}(q) = 1 - f_{fc}(\vec{s}_{1..M}(q)) \tag{5.5}$$

Function f_{fc} is the force closure measure defined by Miller and Allen [1999]. This measure uses the Coulomb friction model to compute the Grasp Wrench Space using the Minkowski sum of each of the wrenches caused by the possible contacts, weighted by the potential. This value is subtracted from one so as to maximise the force closure by minimising this function. $\phi_{fc}(q)$ assumes that the projected points will become contact points when distance to surface of the object decreases, which is valid when close to the object. AICO allows for dynamic weighting of the costs across time, and so the weighting for ϕ_{fc} is set to increase from zero to the target weight as the hand approaches the object. This allows the cost function to increasingly consider the local geometry and final finger tip placement in regards to grasp quality as the hand performs the final grasp.

5.3.3 Hand Envelopment Measurement using Electric Flux

As described earlier in Section 4.4, flux can be used to specify the envelopment of a surface around an object in a manner similar to the coverage of a surface around a

point computed by the winding number. By attaching a flux measurement surface to the inside of the hand which deforms as q changes, a metric for the hand's envelopment around the object is created. The measurement surface T_m is defined by using points p_j as vertices. Since the aim when power-grasping an object is to maximise the area that envelops the object, the surface is defined such that it spans the whole hand and connects the areas between fingers in the human hand model. This leads to the spreading of the fingers across the surface of the object when maximised, such as during power grasping motions, which leads to more stable grasps. The state term for flux in a certain configuration is given as:

$$\phi_f(q) = \Phi_{\text{Flux}}(T_m, T_{\text{obj}}, Q_{\text{obj}}) \tag{5.6}$$

The dimensionality of electric flux space can be chosen arbitrarily anywhere between 1 and D dimensional space, where D is the number of triangles comprising the flux measurement surface, by grouping triangles. Lower dimensional space allows for simple control while higher dimensional flux spaces allows us to deform parts of flux measurement surface independently. This is crucial when controlling style of grasping where each finger moves differently based on the style. Control over the style is possible by weighting each triangle separately. These weights can either be defined manually by an expert or they can be learnt from demonstration.

5.4 Motion planning

Given the abstract representations introduced in Section 5.3, the user would wish to compute the trajectory in the joint space $q_{0:T}$ and the controls $u_{0:T}$ to execute this trajectory to perform a reaching and grasping motion using a robotic manipulator, where T is the number of frames to look ahead in the motion. This is called motion planning. Due to the simplicity of the space of the Electric Parameters, a complex planning algorithm is not required. A valid choice for planning in the above defined abstract spaces is instead the Approximate Inference Control framework (AICO)[Toussaint, 2009] for stochastic optimal control. It seeks to find the trajectory of the robot controls $u_{0:T}$ where the task cost in minimised despite the presence of noise in the readings of the robot configuration or in the systems underlying the cost definition. AICO has recently been used in robotics by Zarubin et al. [2012] to successfully plan in topological spaces (such as the writhe space), which have similarities to the Electric Parameters, and also

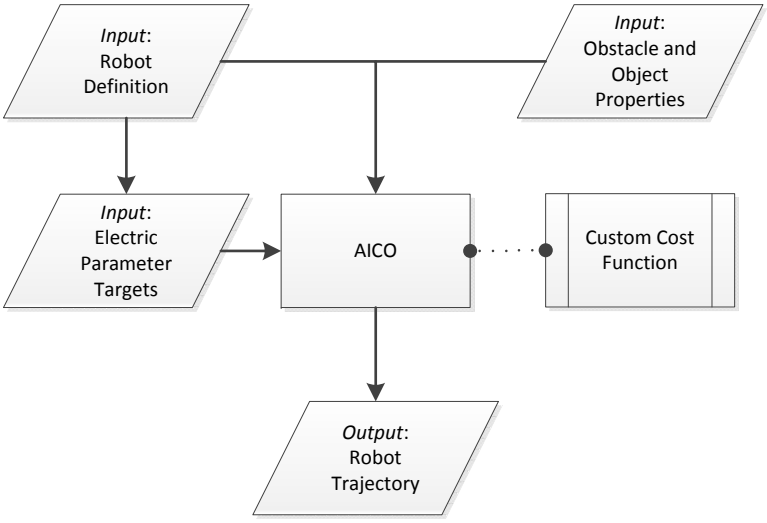


Figure 5.5: A block diagram showing the relationship between the input, AICO, and output. The custom cost function is shown in Equation 5.7.

in time-varying reaching tasks by Rawlik et al. [2013], demonstrating its suitability for this work.

5.4.1 Computing Novel Trajectories using the Electric Parameters and AICO

AICO is used for this chapter's purpose as shown by the flowchart in Fig. 5.5, in which the robot motion is made to match the original electric parameters captured from the human as closely as possible whilst the motion remains physically achievable by the robot, and given constraints such as force closure and control cost which is added by the cost function and inference process. AICO essentially aims to minimize the squared distance between the desired state y_t (e.g. in the electric coordinate space) and the current state computed from the joint configuration q_t (as defined in Equation 5.2 for electric potential and Equation 5.4 for projected UV), given specified soft constraints and weighted by a precision constant ρ. Note that using a local linearisation of ϕ (having the Jacobian of the Electric Parameters and other costs) is sufficient[7]. Here the assumption is made that trajectories in these task spaces can be effectively represented using multivariate Gaussian distributions, which is justified by the fact the electric field is homogeneous and it contains no local minima. The task cost for state x

[7]This can be computed either analytically or numerically using the GPU.

at time t is defined as

$$\begin{aligned} c_x(x_t) = &\rho_p \|\phi_p(q_t) - y_p\|^2 + \rho_{uv} \|\phi_{uv}(q_t) - y_{uv}\|^2 \\ &+ \rho_f \|\phi_f(q_t) - y_f\|^2 + \rho_{fc} \|\phi_{fc}(q_t)\|^2 \\ &+ \rho_c \|\phi_c(q_t)\|^2 \end{aligned} \tag{5.7}$$

where $\phi_p(q_t)$, $\phi_{uv}(q_t)$, $\phi_{fc}(q_t)$ and $\phi_f(q_t)$ are defined in equations 5.2, 5.4, 5.5 and 5.6 respectively, and $\phi_c(q_t)$ is the collision cost computed as reciprocal distance to the closest obstacle. The various ρ factors are weightings per element which are set by the user. By default they are set to the values in Table 5.1, which were determined by informal tuning from initial relative importance values. We consider ρ_{uv} to be more important than ρ_p as the location on the object surface is of greater impact than the distance to the object surface, especially during the approach, and at the final stage the fingers are closed into contact with the object. Force closure and flux are weighted lower as to subtly direct the hand when target positions are given and adjust to new situations. Finally, the reciprocal distance to contact is also weighted lower as to prevent penalising close interaction when it is requested by the electric coordinates.. The Jacobian of electric flux and the collision measure are computed via the chain rule (see Section A.4 in the appendix for the electric flux Jacobian), and the finite differences method is used to approximate the Jacobian of the electric coordinates and force closure. The control cost is implicitly added through the inference process. This is then solved using the AICO framework to produce a trajectory with (locally) minimal cost overall.

Weighting Term	Value
ρ_p	10^{-1}
ρ_{uv}	1
ρ_{fc}	10^{-3}
ρ_{flux}	10^{-2}
ρ_c	10^{-3}

Table 5.1: Weights used in the cost function during the experiments, unless otherwise specified.

Manipulation of the Object The representation of motion shown here only provides the relative motion of the hand to the object. This means that using this representation

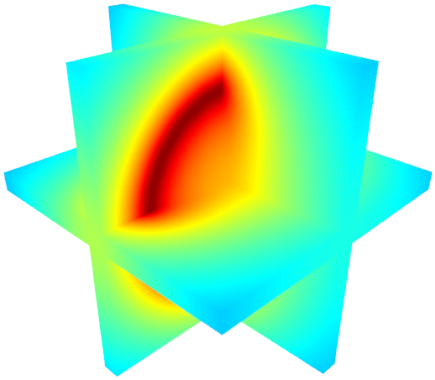


Figure 5.6: Slices of the precomputed electric coordinates for a torus. The potential component is displayed here in colour, where red is high potential and blue is low.

directly it is not possible to move the object accurately. In the case of manipulation, once the object is grasped by the hand, the object can be considered an extension of the wrist instead. The path the object must travel is known from the original motion data.

This allows for the target location at each frame for the wrist to be computed by the inverse transform of the wrist to the object. Inverse Kinematics can then be used to move the wrist to the target location for each frame using traditional methods on the configuration q of the robot.

5.5 Implementation

This section describes the precomputation and adaptation of the field so the electric coordinates can be looked up at runtime and is can be used in the internal space of the object, as well as the external space.

5.5.1 Precomputation

Although the computation of the potential and UV for hand points $\vec{p}_i(q)$ can occur at run-time, the electric coordinates at the vertices of a 3D grid structure surrounding the object are pre-computed to decrease run-time computation and therefore speed-up the grasp generation (Fig. 5.6). The potential and UV for any point in space can be computed by trilinear interpolation of the neighbouring vertices when inside the grid,

and extrapolation outside the grid. The grid's location is then transformed based on the object transform during runtime, allowing for look-up of the electric coordinates (ϕ_{uv} and ϕ_p) for points $\vec{p}_i(q)$ in the grid at each frame.

For the hand the electric potential computational complexity is $O(L)$ for a single sample[8] (where L is the number of triangles comprising the object), and $O(LM)$ considering all the attached points, where M is the number of points attached to the hand as defined in Section 5.2. This lookup grid reduces the computational complexity of the potential calculation for a single point from $O(L)$ to an $O(1)$ lookup and trilinear interpolation. It also reduces the time taken to compute the corresponding point on the surface of the object given a point in space, as this is performed by using numerical methods of integration along the field line until the surface is reached during precomputation. This means that using the lookup grid ϕ_{uv} and ϕ_p are calculated during runtime in $O(M)$ time, eliminating the reliance on the complexity of the object mesh.

The desired density of the grid depends on the surface of the object: for one that has high-frequency variation and denser grid must be used. In the experiments in this chapter, a grid of 100 points per axis were used, which spans the bounding cube of the model.

5.5.2 Combining The Electric Field with the Interior Distance Field

The calculations presented as an approximation of the electric field thus far are only valid for the outer space of the object. Although theoretically the field is a consistent 1 volt both on and inside the surface of the object in the physical interpretation of the field, sampling our approximation of this field inside the object leads to values less than the surface value of 1 volt. This not only causes a local maxima at the surface of the object but means that the local potential value cannot be used to define whether a point is 'inside' the object or not, meaning a separate collision detection phase has to occur for what may be a complex object.

To alleviate this problem the computed voxel grid is post-processed to store the values of a distance field as the potential for interior points, which computes

$$1 + ||\vec{x} - \vec{b}|| \tag{5.8}$$

where $\vec{x}$ is the sample point in question and $\vec{b}$ is the closest surface point. In this way

[8] This is defined by the superposition principle and discussed in the representation chapter (Chapter 4).

the field increases in the direction of the medial axis of the object and decreases in the direction of the nearest surface. A point that finds itself inside the object is able to escape the object by moving along this field towards a potential value of 1, bringing it to the closest point on the surface of the object.

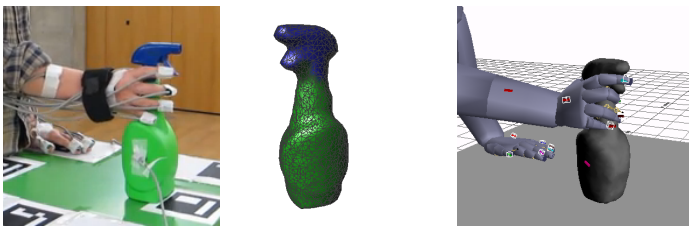
In practice this is computed by using the Bullet physics engine[9] to emit a ray in an arbitrary direction from the sample point, which returns the first point on the object that is hit. If no object is hit, the point can be safely ignored as it is outside of the object (with the assumption that the object geometry is closed). If the object is hit, the dot product between the ray and the normal vector for the triangle that was hit is computed. If it is positive (the ray has hit on the inside of the triangle), then the potential of this point is recalculated using Equation 5.8.

5.6 Experiments

To validate the proposed technique experiments were carried out for motion transferral and planning using the Electric Parameters, specifically focusing on object approach and grasping. The KUKA LWR4 robotic arm was used in combination with the Schunk Dexterous Hand (SDH servo-electric 3-Finger Gripping Hand) [SCHUNK GmbH and Co, 2012], the Shadow Dexterous Hand [Reichel and The Shadow Robot Company, 2004] and the KCL Metamorphic Hand [Wei et al., 2011a] in experiments to show transferral of human motion to robotic hands with different capabilities. These manipulation platforms have 14, 31, and 23 DoF respectively, including the arm. These hands were chosen as they cover a range of situations, respectively: a common dexterous industrial robotic hand with a differing number of fingers from the human, a hand designed to be as anthropomorphic as possible, and a hand which has significantly different morphology.

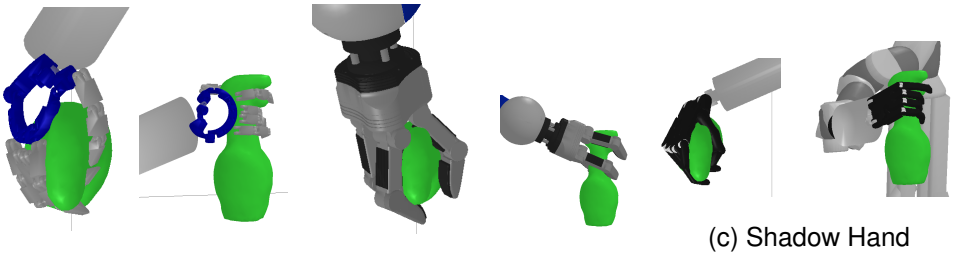
This section begins by showing experiments of motion transfer from a human example of interaction to the three robots in simulation. These simulations are validated by successful execution on the physical hardware of the Schunk hand. Then, in Section 5.6.1.1, experiments are performed which demonstrate transferral of grasps between objects in various locations using the same manipulator. These show that preserving the spatial relationship also preserves the natural approach and stability of a grasp when the objects are of the same class, even when the topology differs. Finally, novel motion planning is demonstrated for the avoidance of obstacles in an environ-

[9] https://github.com/bulletphysics/bullet3



(a) (b) (c)

Figure 5.7: The final frame of a captured example motion of grasping a spray bottle. (a) Performing the motion with the markers attached. (b) The captured geometry of the object using the Kinect. (c) The grasp in simulation.



(a) KCL Hand (b) Schunk Hand (c) Shadow Hand

Figure 5.8: Spray grasp transfer from the human (Fig. 5.7) to three dexterous robotic manipulators: the KCL Metamorphic Hand, the Schunk Dexterous Hand, and the Shadow Dexterous Hand. Transferred motion is applicable to different robot hand morphologies.

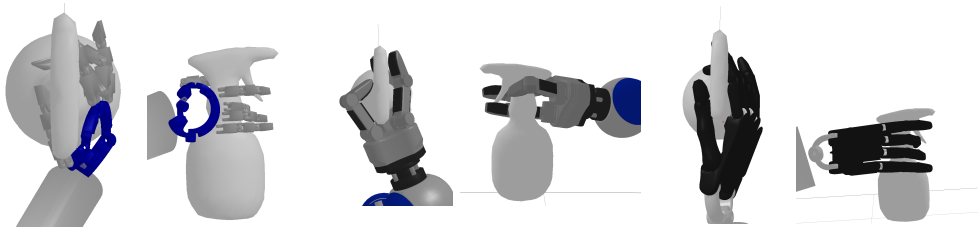
ment whilst preserving grasp style. Evaluation of each of the generated motions is performed by computation of the force closure metric on the grasp pose (to evaluate grasp stability) and visual evaluation of their semantic similarity.

5.6.1 Motion Transfer

The following experiments deal with the cases of motion transfer from human to robotic manipulator, and motion transfer between different objects by a single manipulator.

5.6.1.1 Transferral from Human to Robot

This section demonstrates a series of grasp transfers between a human hand and a robotic hand using various grasps on objects from the database: a spray bottle, a small



(a) KCL Hand (b) Schunk Hand (c) Shadow Hand

Figure 5.9: The final grasp poses of a grasp transfer from the human (Fig. 5.7) to the robots with a novel spray bottle.

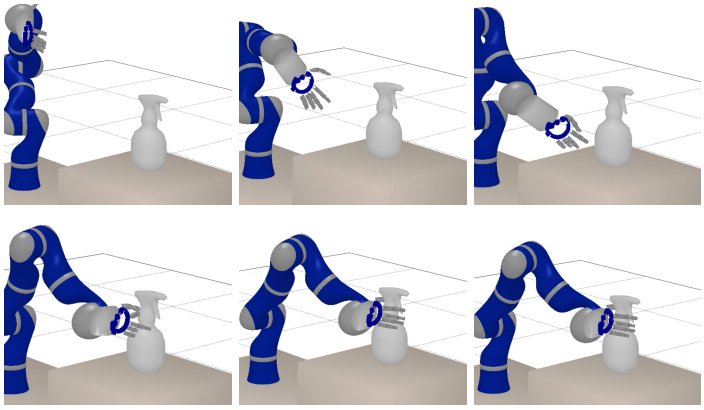


Figure 5.10: Sequence of motion transfer to a novel spray bottle grasp, initially captured from a human, to the KCL Metamorphic Hand. The hand interpolates in the electric coordinates to achieve the start configuration, then approaches as the human does.

drinks bottle, a teddy, and a bowl. These are transferred to novel objects and to other manipulators.

An important part of the transferral of motion is capturing the original scene accurately, as the transferral process relies heavily on the initial motion. Interactions from the database captured using the technique described in Chapter 3 are utilised here. The technique presented in Chapter 3 enables the capture of the close interaction of the actor and the environment. This capture results in an digitized scene with both object geometry and motion data for the actor and object (see Fig. 5.7 for an example).

Setup Firstly, hand joint positions are selected and their electric coordinates are recorded during playback of the motion as described in Section 5.3.1. During the experiments, it was established that selecting the finger tips, interphalangeal joints, and the wrist gives a good representation of the whole hand motion. This gives a per-frame

Grasp Type	Schunk FC	KCL FC	Shadow FC
Spray Trigger	0.0579	0.0676	0.160
Spray Base	0.0523	0.0428	0.0571

Table 5.2: The force closure values for different grasps on the novel spray bottle using multiple manipulators. As is usual, the Q value for force closure is normalised and so does not require units.

sequence of points in the space of electric coordinates. In addition to this, the total flux passing through the flux measurement surface defined over the inside of the hand is also computed, thus providing a measure of coverage and overall orientation towards the object.

Secondly, a mapping between the fingers and the robot hand is given by the user, stating which robot finger corresponds to which human finger. The electric coordinate representation of the motion previously captured can then be used to direct the robotic arm using the AICO framework. Task variables are set to: electrostatic flux difference $\|\phi_f(q_t)-y_f\|^2$ (the flux passing through the triangles shown in Fig. 5.4), electrostatic potential difference $\|\phi_p(q_t)-y_p\|^2$ and electrostatic UV projection difference $\|\phi_{uv}(q_t)-y_{uv}\|^2$ with the weighting as shown in Table 5.1. The three robots have rubber-like pads on the inside of the finger tips, and so a friction coefficient of 0.5 was chosen to reflect this. This is a low estimate for the majority of objects that humans interact with in everyday situations, but an underestimate helps to ensure that the grasp is stable in real-world applications as well as in simulation. As the robot pose cannot be assumed to match the human's initial pose in the examples, a small number of frames are added as interpolated electric coordinate targets for the initial pose of the robot to the initial coordinates in the example motion.

Optimisation over the cost function is then performed using AICO, which minimises the difference in the electric coordinates between the original motion given by the human hand and the newly generated motion for the robot hand and arm. For the final frame, if the fingers that were close to the object were not in contact (had high potential but were not at 1 volt) we increased the flexion of each until there was contact, or until a joint limit was reached. By transferring the interaction to Shadow Dexterous Hand, KCL Metamorphic Hand and Schunk Robot Arm, the method developed in this thesis is shown to be generally applicable over robots of dissimilar kinematics.

Grasping a Spray Bottle The first transfer is of a spray bottle, which is a tool with a standard grasp for use as shown in Fig. 5.7(a): the fingers wrap around the bottle neck and one or more fingers placed on the trigger. This was chose as it is a complex grasp which cannot be inferred using geometric properties alone: knowledge of the tool must be used to choose the correct grasp for use. This is denoted as the "Trigger" grasp. A second capture in which the actor grasps the lower half of the spray bottle is also used, which is denoted the "Base" grasp. This is a grasp that may be used for transporting the object.

The initial transfer used the original scanned spray bottle model for transfer. All grasps were semantically similar and successful, as seen in Fig. 5.8. The grasp can also be transferred to novel objects, and so a second spray bottle model was targeted.

In order to transfer the same grasp style between objects, the input motion is mapped to the electric coordinates, which is then mapped between objects by creation of the smooth surface correspondence between the two using Blended Intrinsic Maps and the isosurface of the electric potential field as described in Section 4.5.2.2. The grasp may then be applied to a new object using the task variables specified above, similarly to the transfer between manipulators.

The transfer of these two spray grasps was successful on all three hand models in simulation (see Fig. 5.9 for the final *trigger* poses and Fig. 5.10 for the full sequence for a single manipulator). There were no self-collisions and, as is evident from Table 5.2, positive force closure was achieved (reported as *Spray Trigger* and *Spray Base*), which indicates that the grasp can withstand some magnitude of wrench in any component direction. The results were visually similar and preserved the style of grasp. As the resulting motion is semantically similar in terms of relative final grasp locations and approach to the object, kinematic tasks can be demonstrated using a natural interface such as our own body even when the target manipulator is significantly different. Applying this technique to the hardware validated this evaluation, showing success for both grasp types, as seen in Fig. 5.11.

Transferral to Multiple Robots and Objects To show generality of the technique to other objects, further captured motions were transferred from a human actor to the three robotic hands, and to novel object shapes in the same class of object. Four captured grasp-and-pick motions using a 500ml 'Ribena' drinks bottle were transferred to a second small drinks bottle and a much larger 1 litre bottle. These motions ranged from grasping at the cap to reaching around the side of the object, to picking the object

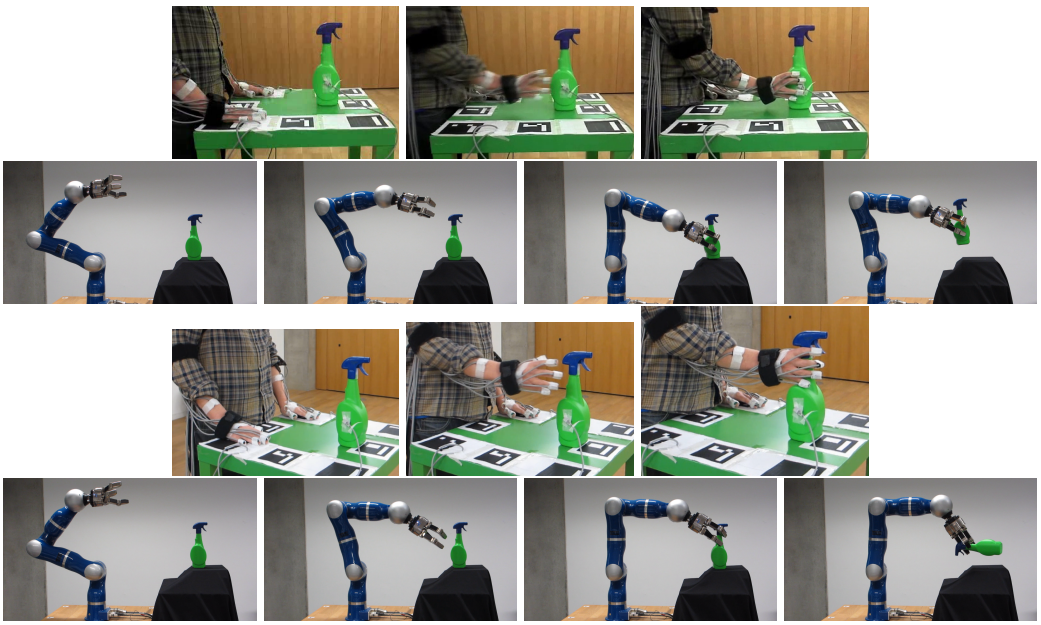


Figure 5.11: Transferred grasps of the spray bottle from the human to the Schunk hand on the hardware. The location and style of grasp is spatially similar although the kinematics are significantly different.

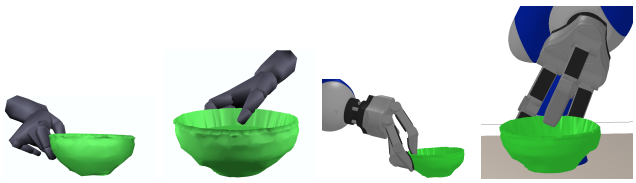


Figure 5.12: A transferred grasp on the edge of a bowl from the human to the Schunk robot.

up from a flat starting location. Results are shown in Table 5.3. When using this technique the fingers adapt to the new geometry and no self-collisions occur. The grasps were robust to placement of the object including orientation. When executed on the simulator, the grasps were validated as stable in all cases aside from when grasping the Ribena bottle at the top with the Schunk hand, as the bottle was too small in comparison to the Schunk hand. This was reinforced when applied to the Schunk hardware as shown in Fig. 5.13, where the grasps shown to have force closure in the simulator were successful. A further capture example of grasping on the side of a bowl yielded interesting results, as the interaction produced a very different pose on the Schunk robot as compared to the human, but the semantic similarity of the grasp is preserved (Fig. 5.12).

In simulation a variety of further objects were used to demonstrate grasp transfer

Grasp Type	Schunk FC	KCL FC	Shadow FC
Ribena Side	0.0104	0.0277	0.0480
Ribena Front	0.0185	0.0254	0.0342
Ribena Top	0 (failed)	0.00142	0.00158
Ribena Flat	0.00169	0.0735	0.178
Lucozade (500ml) Side	0.110	0.0107	0.00754
Lucozade (500ml) Front	0.00527	0.0125	0.0854
Lucozade (500ml) Top	0.0389	0.00254	0.00189
Lucozade (500ml) Flat	0.00736	0.000521	0.0182
Lucozade (1L) Side	0.173	0.0474	0.00499
Lucozade (1L) Front	0.0784	0.0322	0.0854
Lucozade (1L) Top	0.0234	0.0288	0 0.00497
Lucozade (1L) Flat	0.00580	0.0496	0.00579

Table 5.3: A quantitative analysis of the force closure values of grasp transfer between the differing bottles, grasp styles and robot hands.

between objects: a soda can, four teddy bears in different poses, a bowl, and a bin with multiple holes, as well as a selection of items from the capture database from Chapter 3. As before, each of these objects are simulated as rigid-body objects.

A transfer experiment performed in simulation between the Ribena drinking bottle and a soda can shows preservation of grasp style between object class. The final grasp pose of the Schunk hand with a drinking bottle is set by the user and captured in the electric parameters. The object is then replaced with a soda can and the previous coordinates are set as targets for local optimisation using AICO. The object was successfully grasped in a semantically similar way without collision, and had a force closure value greater than zero, denoting the grasp is stable (Fig. 5.14 and Fig. 5.15).

Teddy Model	Schunk FC
163	0.0181
164	0.0111
165	0.00457
166	0.000372

Table 5.4: The force closure values for a transferred grasp on the arm of teddy-163 to the other teddy models.

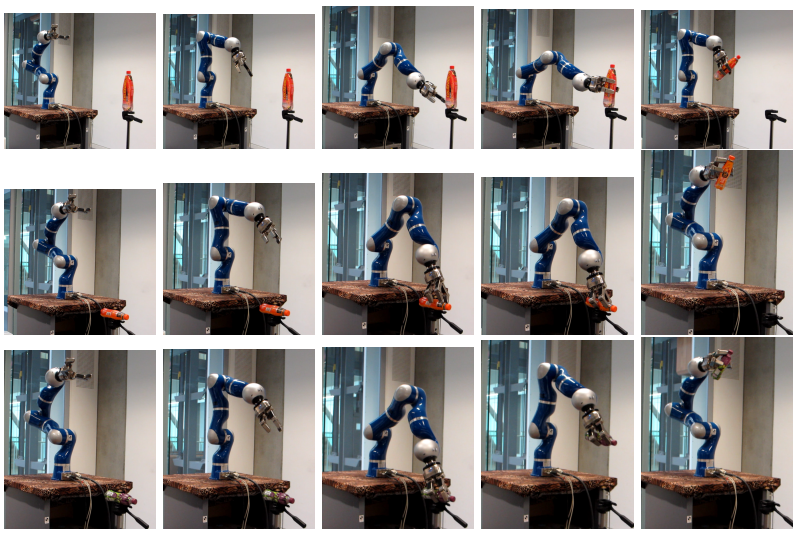


Figure 5.13: Example grasps by the Schunk hand on various bottles in different configurations, transferred from a single human grasp. Note that as well as variation in object geometry the technique is robust to the location and rotation of the object as we define the target relationship between the manipulator and the object.

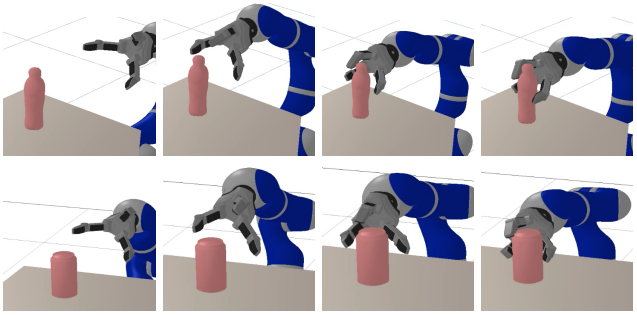
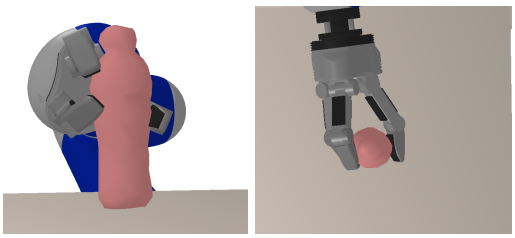
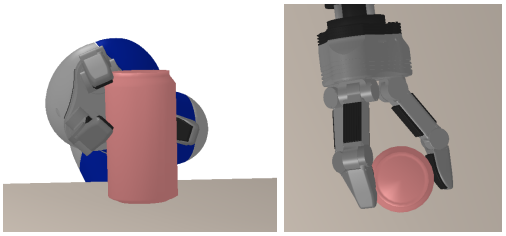


Figure 5.14: Sequence of motion transfer of grasping a 500ml bottle to a drinks can. The motion was defined solely by the electric parameters of the final pose. Note the finger passing over the top of the bottle.

Transfer experiments that demonstrate the preservation of the grasp style have also been performed on an object in different poses. These experiments were performed between the different teddy bear models in our shape database, and show the successful mapping and grasp transfer of an articulated object. An initial grasp is defined on the arm of the teddy in the first pose. Three more poses are defined, and the correspondence is computed. The UV coordinates are mapped using this correspondence to the differently posed models and these are combined with the original potentials and set as final electric coordinate targets. All transferred grasps are stable and applied on

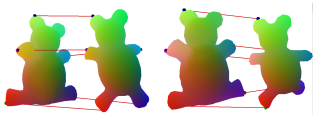


(a) Initial bottle grasp (front and top view)

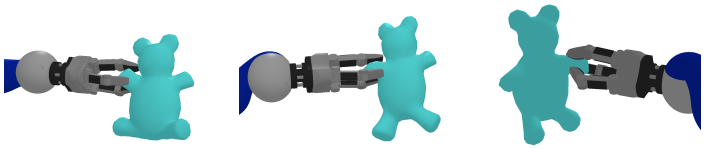


(b) Transferred grasp to soda can (front and top view)

Figure 5.15: The final grasp from the sequence shown in Fig. 5.14. These two objects are different but the grasp can be transferred without any tuning of parameters.



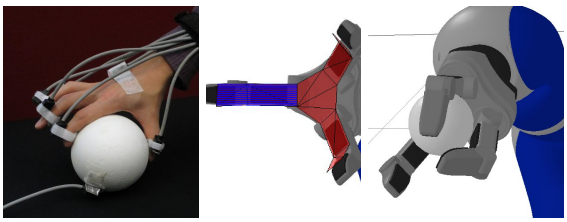
(a)



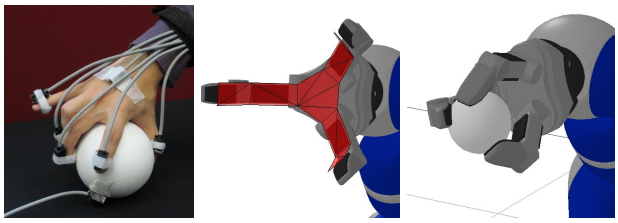
(b)

Figure 5.16: We transfer projected points between objects by using a correspondence between the object surface (a). Using Blended Intrinsic Maps creates a non-rigid correspondence suitable for transfer when objects are similar topologically but differ in subpart pose. (b) shows a grasp on the arm of the bear transferred to three bears in different poses.

the arm of the teddy, as shown in Fig. 5.16 and Table 5.4. In certain situations, our correspondence function mirrors the true correspondence, which is a known limitation of Blended Intrinsic Maps[Kim et al., 2011]. This can be seen in Fig. 5.16(b), final image, in that the teddy's left arm has been mapped to the right. Alternative mapping



(a) Two finger grasp



(b) Three finger grasp

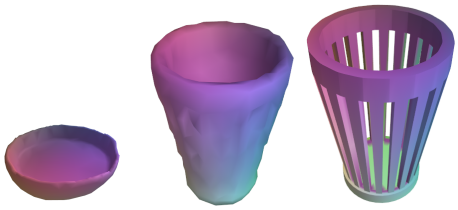
Figure 5.17: Grasping objects using electric flux. A two finger grasp (a) and three finger grasp (b) achieved through a weighting which is measured from an example grasp. An illustration of the weightings are shown on the flux surfaces here. Blue striped surfaces have lower weights. The motion is semantically similar and adapts gracefully to new relative positioning of the hand and the object.

functions may be used in place of this to overcome this limitation.

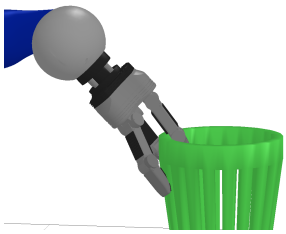
Transfer to Objects of Different Topology It is also possible to demonstrate grasp transfer when the objects are topologically different. An initial grasp on the lip of the bowl (Fig. 5.12) was transferred to the bin with topological holes, as shown in Fig. 5.18. Although the bin is significantly different, the mapping is successfully computed using the correspondence technique described in Chapter 4. The stable grasp is then transferred to the edge of the bin by transfer of the electric coordinates.

5.6.2 Novel Motion Planning

The previous section shows motion transfer when the object or manipulator changes. This section focuses on the introduction of an obstacle into the environment. Offline motion planning is performed in the electric parameter space for purpose of grasping the spray bottle with this obstacle present (see Fig. 5.19). The system is set up with the following task variables in addition to the electric coordinates and flux: collision avoidance $\|\phi_c(q_t)\|^2$ and force closure of the electrostatic projection of the finger tip



(a)

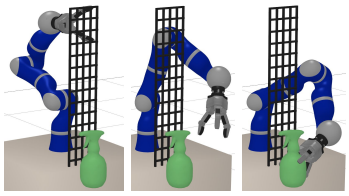


(b)

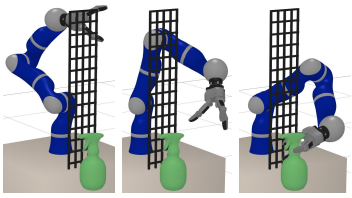
Figure 5.18: The bowl grasp from Fig. 5.12 transferred to a topologically different model. This bin has many holes, so we use the isosurface at 0.99 volts (left) to generate a genus zero mesh that closely resembles the original model (right). We can find the correspondence between this genus zero isosurface and the original bowl using Blended Intrinsic Maps and transfer the target UVs in this way (a). (b) shows the resulting grasp on the original model.

positions $\|\phi_{fc}(q_t)\|^2$ (for a stable grasp). The original transferred trajectory is used for initialisation. Since all of the task variables are well defined everywhere in the workspace, we are able to iterate AICO until convergence. Fig. 5.19 shows the result of planning a reaching and grasping interaction for a scenario where naive motion transfer fails, as there is an obstacle in the way. The planner computes a stable generic grasp for the spray bottle when using flux maximisation alone(see Fig. 5.19a). By adding target task variable using the UV coordinates similar to the transfer experiments, the grasp can be directed to certain locations on the object. In this case, however, a target in UV coordinates is specified only at the final time step of the trajectory. It is then possible to to encode a bias (or a style) while the remainder of the trajectory relies predominantly on flux and force closure priors. Fig. 5.19b shows the resulting precision grasp around the neck of the spray bottle and Fig. 5.19c shows the robot executing this plan and subsequently picking up the spray bottle.

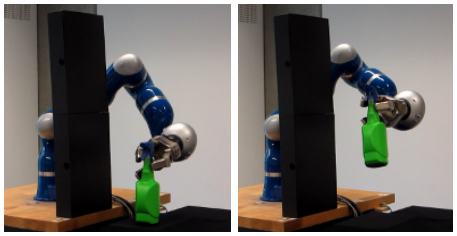
Electric Flux Weighting Electric flux alone can also be used alone in guiding the hand to grasp, without an explicit trajectory to follow but with a certain style. In



(a) Force closure grasp (ϕ_{fc}: 0.0456)

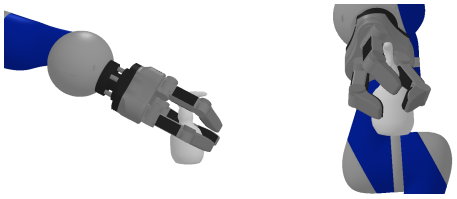


(b) Precision grasp (ϕ_{fc}: 0.0322)

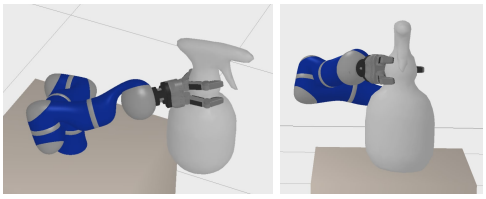


(c) Precision grasp and pick up

Figure 5.19: Planning successful grasps while reaching around an obstacle. (a) Force-closure grasp maximising stability. (b) Precision grasp with style defined using electric coordinates. (c) Precision grasp implemented on hardware and picking up the spray bottle.



(a) $0.5\times$.



(b) $2\times$.

Figure 5.20: The trigger grasp transferred to a $0.5\times$ size spray bottle, and a $2\times$ size spray bottle.

order to demonstrate transfer of different grasp styles using electric flux without the electric coordinates, simplified experiments are performed on a sphere. By setting the task variables to only consider flux per finger, two finger and three finger grasp styles are able to be produced by demonstration. Flux is suitable to be used solely when the specific location to grasp on the object is not important. Results are shown in Fig. 5.17a and Fig. 5.17b. These grasps are stable in simulation and when applied to the hardware.

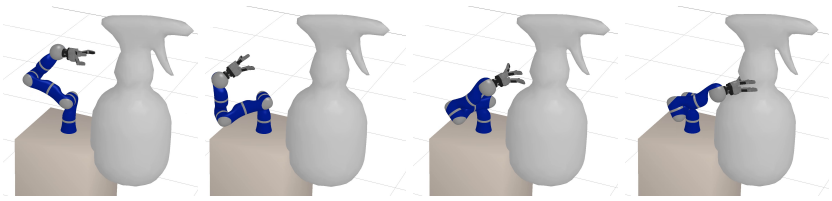


Figure 5.21: The trigger approach and grasp transferred to a 5× size spray bottle. This is not a successful grasp as it is not force closed, and does not preserve the use of the tool.

5.6.3 Limitations

There are some limitations to the technique. As the method relies on preserving the spatial relationship, when the hand is much larger or smaller relative to the object in the demonstrated grasp, as compared to the target situation, the semantic similarity of the grasp cannot be preserved. In figure Fig. 5.20, the trigger grasp is transferred to a very large and a very small bottle. Although the grasps are both stable, no finger is able to reach the trigger, thus rendering the tool unusable. In the extreme case of a 5× size spray bottle shown in Fig. 5.21, the approach requires the hand to move far backwards to preserve the original approach, and the grasp is not stable. In these cases, the example grasp can be considered poor, but it means that the user must provide examples on objects that do not differ too greatly to ensure that the system can produce the required grasp with high quality.

The technique also relies on global correspondence generation between the source and target object. Although this can always be generated, it may be unintuitive from a human perspective. For instance, in Fig. 5.16(b) the teddy in its initial pose is grasped by its right hand, but when mapped to a differently posed teddy the mapping flips the right and left arms due to near-symmetry in the geometry. As our technique is agnostic to the specific mapping, requiring only a dense conformal mapping to be given, this can be mitigated by future improvements in dense correspondence techniques.

Finally, planning for grasps that pass through handles is slow (in comparison to treating the object as genus zero) as the UV distance metric is then required to take the object's topology into account. This is done by using the geodesic distance from the current UV and the target UV on the object's surface, but is expensive to compute. This means that only grasps that do not pass through handles can be transferred in an online fashion. In future work this may be able to be mitigated by considering the topological graph of the open space of the object (for example by using the singularities detected

in Section 4.5.2.3 or the Reeb graph [Hilaga et al., 2001]), and computing the UV distance using this simplified representation, but this requires more investigation.

5.7 Conclusion

This chapter exploited a novel representation for grasp transfer based on electrostatics to parametrise the salient aspects of the demonstrated grasp in terms of spatial relationship by using the flux as an envelopment measure and the electric coordinates of multiple points per link as the relative position and orientation.

The framework presented in this chapter aims to reduce the influence of small scale variations in local object geometry, and allows for the consideration of both the caging and force closure of the grasp by means of appropriate scoring functions. The first scoring function uses electric flux to define the envelopment of the object by the hand, and the second is based on the electric coordinate system that models the projection of a robot hand onto an object. The first function is defined in terms of the *integral* of a locally varying function over a surface defined in terms of a robot hand and hence exhibits a larger degree of stability under variations than scoring functions defined only for point-wise geometric quantities.

The advantages of this electrostatic representation for grasping are multiple: a continuous mapping from the object's surface to the outer space, non-intersecting field lines, a definition of a coverage measure for the entire outer space of the object, and a mapping between arbitrary objects using the isosurface alongside global correspondence techniques. By working in this alternate space that focuses on the relational aspects of the grasp rather than absolute kinematics, and utilising Approximate Inference Control (a local optimisation technique) it is possible to couple the motion in these abstract spaces with trajectories in the configuration space of the robot.

As flux measures the envelopment of the object, maximisation of flux leads to the hand maximally enveloping the object. As previous techniques have noted [Zarubin et al., 2013], there is a correlation between grasp quality and envelopment. The work outlined in this chapter demonstrates an envelopment measure for grasping that can be locally optimised no matter the relative location of the hand and the object.

The experiments conducted show that force closure values tend to be greater on the robot models with five fingers as there are a greater number of contacts, and the similarity to the human hand meant the style of approach and grasp is better preserved, but these manipulations are also able to be transferred to morphologically dissimilar

manipulators such as the KCL hand, which has a dependency loop in its kinematics, and the Schunk dexterous hand which has only three fingers.

This chapter demonstrated that the method computes successful grasps that generalise over objects of different shapes and robots of dissimilar kinematics while visually retaining the qualitative grasp type. The grasps were stable, thus allowing the object to be manipulated. Although the pre-processing stage takes a significant amount of time to generate the field, this chapter also presented optimisations that, after pre-processing, make it possible to use these representations for real-time applications such as teleoperation.

Chapter 6

Conclusion

6.1 Overall Contributions

This thesis has shown contributions in three main areas that together make up a novel technique for capture and generalisation of close interaction with objects for transfer of close interactions. A specific emphasis is put on the naturalness of the approach as well as the final grasp configuration. The technique allows for programming robotic interactions with objects by human demonstration on dissimilar objects.

Firstly I proposed a novel capture technique that captured interactions with objects without occlusion, and also captured the geometry for use in computation of interaction representations that require geometry. An RGB-D sensor combined with visual markers is able to capture the object geometry, and the alignment with magnetic markers placed at landmarks in the scene allows for the marker offset to be automatically computed. A technique for reducing error caused by eddy currents in the magnetic field is shown, making the system practical for capture of close interactions. These magnetic markers are shown to be accurate in respect to relative positioning using this technique. A publicly-available database of interactions was created using this technique.

Secondly, a parametrisation of motion called the Electric Parameters is presented, allowing a representation of interaction independent of specific geometry and topology. It combines the electric coordinates with electric flux. Electrostatic flux is a measure of envelopment created using a virtual electric field, shown in this thesis to be useful in caging grasps and envelopment motions. This harmonic field can be used to define a parametrisation across the whole space with respect to the object surface called the electric coordinates. Relative position and orientation can be encoded by them. Grasps can be demonstrated by a user on an object and transferred to novel objects and novel

manipulators in a physically valid way using this technique. This is achieved via an extension to Blended Intrinsic Maps to create a semantically valid dense conformal map between objects with holes or disconnected components utilising the genus zero isosurface of the electric field.

Finally the thesis demonstrates robotic manipulation transfer and generation by the combination of the interaction capture technique and electrostatic representation defined earlier, along with local optimal control methods. These local methods allow for teleoperation, and programming by demonstration given an example interaction, to enable the robot to imitate an object manipulation on dissimilar objects.

6.2 The Bigger Picture

This thesis deals with a number of different techniques in different fields, but mainly considers the problem of representation of spatial relationships. In motion capture, this was considered in the true capture of these relationships, avoiding pitfalls caused by estimation and occlusion. For animation and robotics a novel envelopment metric was defined that can be used for any enveloping relationship in the outer space of the object. The general relationship of points in space relative to arbitrary objects was also defined, allowing for an alternative to relative vector sampling on the surface and object centroid relationships. Additionally, transferral of these relationships was investigated by a novel technique for dense correspondence between arbitrary objects. The spatial relationship was then preserved in order to transfer the motion to new manipulators and objects.

The ability to capture the close interaction with objects is something that the state-of-the-art techniques for capture struggle with. Wang et al. [2013b] uses multiple cameras to attempt to avoid occlusion and both Hamer et al. [2011] and Oikonomidis et al. [2011a] use inference based on the previous frame and partially occluded hand to estimate the pose of the hand when the data isn't directly available. Optical techniques such as that of Huang et al. [2011] also have problems of occlusion. The technique in this thesis avoids the issue of missing data by using sensors that are not possible to occlude optically. Previous alternatives to visual or optical capture such as that of Roetenberg et al. [2013] or Vlasic et al. [2007] also attempt to prevent missing data but in doing so lose the ability to obtain the global transformation of a free-floating marker, such as one that may be attached to an independent object. This is not the case with magnetic systems, and this thesis combines this with capture of the object's surface as

they have not been used before with automatically captured object geometry, but have been for capturing dexterous motion [Mitobe et al., 2006].

Much of the previous research relating to transfer of motion with objects focuses only on the grasp, and does not consider the naturalness of the approach. Other motions relative to the object but not in contact can be important, such as the placement of the finger near a trigger, but these were also often ignored. By using the representation presented in this thesis, these motions can be encoded clearly and easily. Recent representations of relative motion in robotics and animation, such as [Al-Asqhar et al., 2013] and [Ivan et al., 2013], either do not handle relationships with concave objects or sample individual points from which relationships are defined, limiting the representation of the relationship to be discrete and meaning parameters have to be tuned regarding the amount of sampling to be performed. Similarly to the goal of Zarubin et al. [2013], the relationship presented here can be used to define a topological invariant: envelopment. However, the work of Zarubin et al. [2013] can not be used away from the object surface, and relies on global search to find good grasp points. Conversely the representation here is useful in the entire outer space of the object and can be used locally for envelopment of an object, which may prove useful for universal grippers [Brown et al., 2010] which do not have fingers. Further to this, the object-centric coordinate system defined by the electric coordinates is a good replacement for techniques that use the distance field to map points in space to an object, as a smooth change in position in the outer space of the object relates to a smooth change on the object's surface. This could improve sampling-based motion generation techniques such as Contact-Invariant Optimisation [Mordatch and Todorov, 2012] and Wang et al. [2013b]'s retargeting approach, as samples would be able to smoothly move across the surface of the object rather than jumping between convex areas.

Representing motion using the Electric Coordinates has another benefit: the ability to transfer the target locations to other objects via surface mapping. Unlike the work of Tegin et al. [2009] or Hillenbrand and Roa [2012], this allows for imitation learning of the entire reach and grasp, and the mapping allows for differences in topology. Limitations of finger pose such as that of Gioioso et al. [2013] are avoided also.

Techniques which use only the traditional object local coordinate system (often defined as a single transformation matrix for a point at the centroid of an object) may also benefit from this novel coordinate system, as there is the ability to transfer directly an initial guess of the spatial relationship to new objects. Relationships based on the local coordinate system are common in the field of PbD [Billard et al., 2008],

so improvements may be made there when considering objects. The initial guess given by preserving the spatial relationship means that only *local* optimisation techniques are required to refine the relationship, and that exploration/global search can be saved for truly novel situations. As exploration is expensive for high-dimensional systems, transferral of previous planned motions using this technique may lead to a decrease in computational cost.

6.3 Limitations

Although there are a number of advantages to using the techniques this thesis has presented, there are limitations to these techniques.

Regarding motion capture, the technique presented in Chapter 3 has the drawback of losing some precision when wireless sensors are used. For the experiments in that chapter, wired sensors were used instead, limiting the space for capture as compared to optical systems. The magnetic system is also very sensitive to the environment, and so requires care in ensuring no magnetically conductive items enter the capture area. This means that additional work must be performed to set the environment up using this system than when using optical systems, and that the results are often worse when capturing the joint angles of the actor only. However, these limitations are tolerated as it is currently the only system capable of capturing the free motion of objects in without occlusion, which is essential when dealing with close interaction.

There are also limitations to the representation of motion in this thesis. The Electric Parameters are computationally expensive, requiring parallelism and precomputation in order to be feasible to use in near real-time applications. Although the parameters are more descriptive in terms of the spatial relationship than other techniques, this additional cost of computation and complexity of implementation may render them little used outside of the research community.

Another limitation of this thesis is the investigation of transfer is limited to that of individual motions rather than multiple examples of the same task. This prevents generalisation of the task from shared features between the examples, and instead relies on the representation of the task to make assumptions about what we wish to preserve about the source motion, and what is possible to discard. An implicit assumption is that the absolute joint angles of the manipulator are not relevant when performing an interaction. This could be rendered false when considering some artistic movements (e.g. dancing) where the relationship between joint angles may be important. There

are many examples of generalisation from multiple tasks in the field of PbD which are currently applied on a joint-angle representation of motion (for example [Ogawara et al., 2003; Hamer et al., 2010a; Sauser et al., 2012; Kim et al., 2014]) whose principles could be transferred to this work, using the electric coordinates as their parameters and learning the relationship between the object and the manipulator in terms of spatial relationships with the surface of the object.

6.4 Future Work

There are multiple directions for future work stemming from this thesis.

Capture Capture of interaction using the technique presented here (and many other techniques) is currently limited by only allowing rigid or articulated objects. Many objects are instead deformable in the real world, even segments of the human body itself are best represented as deformable, and this leads to penetration when they are not represented as such. By using portable geometry acquisition devices in combination with the accurate capture of the motion of points on deformable objects using markers, the motion of the deformable object can be reproduced in simulation by using the constraints given by the markers along with the noisily estimated state of the deformable object given by the geometry acquisition device. This is interesting as it is currently hard to investigate interaction between humans and deformable objects as there is little digital data, and what data there is often contains significant occlusion.

A further interesting idea is to extend the capture technique described here and use the RGB-D camera during the whole interaction. As the mapping between the visual space and the magnetic marker space is computed, fusion of the sensor readings can be used in order to increase the accuracy of the capture. In this way an initial guess on the object and character configuration could be computed as before using the magnetic sensors, but then refined using visual techniques to estimate the configuration when not occluded. This would give the advantage of non-occluding sensors with the accuracy of visual estimation of the scene. With cheaper and superior RGB-D cameras emerging, this class of technique could lead to highly accurate interaction capture.

Representation One direction for extending the technique for computing the representation proposed here, rather than storing additional information about the scene, is simply to reduce the computation time for the current representation as current pre-

computation is required for interactive applications when using the electric coordinates. Using the GPU and Robin Hood method for computing the charge distribution over the surface of the object has been beneficial (as shown in this thesis), but it is possible more can be done by considering simplified versions of objects initially, and then propagating the charges to the original, more complex versions of the objects for refinement. For example, by grouping the triangles on an object surface, we can consider them to have equal charge and use the techniques described in Section 4.4.3.2 to compute the estimate of the overall charge of the object. This may then be refined by redistributing charge within these groups in parallel in order to compute the true charge distribution over the original object. It may even be possible to have a precomputed database of objects and to choose the one most similar to a novel object to initialise the charges, however it remains to be seen whether this would be faster and would certainly increase storage requirements.

For applications which are not time limited, it may be worth considering an approach which does not consider each triangle as uniformly charged, and rather the charge varies across the triangle surface, for which analytical solutions are known [Tatematsu et al., 2002]. This would lead to high computation requirements (so was ruled out for the purposes of this thesis) but would be more accurate in terms of the isosurfaces of the field.

Manipulation Generation Although manipulation of objects is possible to do using this framework, it is limited to translations and rotations using the full hand. Instead, in-hand manipulation would be an interesting topic to tackle using the techniques shown here. Configuration sampling methods may be required, but generation of motion could possibly be computed quickly by using sampling in the Electric Parameter space rather than the configuration space directly. The representation allows for smooth traversal over the surface of concave objects as well as convex, and so by combining with methods such as Contact-Invariant Optimisation goal-directed motion can be generated by simple target locations for objects in scene on arbitrary objects.

6.5 Concluding Remark

It is clear that there is still much work to be done in the areas of spatial relationship encoding and learning manipulation from demonstration. However this thesis has made progress in these fields and demonstrated transfer of grasping and static manipulation

based on preservation of the spatial relationships in novel situations. It is my hope that this work will inspire others to investigate utilising representations of relative motion, such as the Electric Parameters discussed in this thesis or relationships in Euclidean space.

Appendix A

Computation of the Electric Potential and Field

For completeness the full equations for computation of the electric potential, field and flux are collated here.

In order to compute the electric field we consider a virtually charged object whose surface is represented by a closed triangular mesh T_{obj} of L triangles with per-triangle charge densities $\mathbf{Q}_{\text{obj}}$, a point P or set of points in the outer space of the object (that are attached to the manipulator) and a flux measurement surface T_m that is a mesh of N_m triangles (also attached to the manipulator).

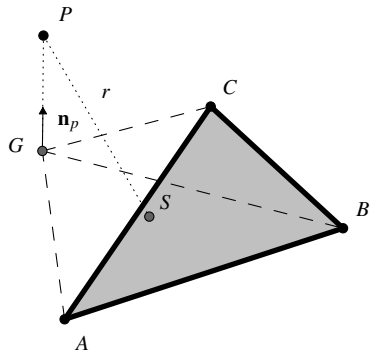


Figure A.1: An illustration of a point P, charged triangle Δ_Q (in grey) and the geometric concepts used for electric potential calculation.

A.1 Electric Potential

Computing electric potential requires a uniformly charged triangle Δ_Q with charge Q and a test point P for which the potential V is calculated. S is defined as the uniformly charged surface of the triangle Δ_Q and $r = |\vec{PS}|$ as the distance from point P to the triangle surface (Fig. A.1);

The potential V is computed as:

$$V = \int_{\Delta_Q} \frac{1}{r} dS \tag{A.1}$$

As in the work of Goto et al. [1992], this is computed by the sum of integrals over three triangles.

Let $\mathbf{n}$ be the triangle unit normal vector defined by the vertices A, B, C of triangle Δ_Q as $\mathbf{n} = \frac{(\vec{AC} \times \vec{AB})}{|(\vec{AC} \times \vec{AB})|}$. Also let G be defined as the projection of point P onto the plane of the triangle such that $G = P - h\mathbf{n}$, where $h = \mathbf{n} \cdot \vec{AP}$. Considering the definition of G, the three triangles that shall be integrated over are defined as Δ_{ABG},Δ_{BCG} and Δ_{CAG}.

To compute the integral for these triangles, a generic triangle Δ_{XYG} is defined, where XY can be replaced with AB, BC, or CA. Let $\sigma = (\vec{XG} \times \vec{YG}) \cdot \mathbf{n}$, which is twice the signed area of Δ_{XYG}. Integration of this generic triangle is computed as:

$$N := \vec{YX} \cdot \vec{YP} + |\vec{YX}||\vec{YP}|$$
$$D := \vec{YX} \cdot \vec{XP} + |\vec{YX}||\vec{XP}|$$

$$I(X,Y,G) = \begin{cases} 0 & \text{when } \sigma = 0, \text{ or } N = 0, \text{ or } D = 0 \\ \dfrac{\sigma \log \frac{N}{D}}{|\vec{YX}|} + |h| \arctan \dfrac{\sigma(\vec{XY} \cdot \vec{XG})(|h| - |\vec{XP}|)}{\sigma^2 |\vec{XP}| + |h|(\vec{XY} \cdot \vec{XG})^2} & \\ \qquad + |h| \arctan \dfrac{\sigma(\vec{YX} \cdot \vec{YG})(|h| - |\vec{YP}|)}{\sigma^2 |\vec{YP}| + |h|(\vec{YX} \cdot \vec{YG})^2} & \text{otherwise} \end{cases} \tag{A.2}$$

The result is zero when σ is zero, as this would be due to a triangle whose normal is orthogonal to $\mathbf{n}$. The result may also be zero when P is coincident with a vertex (i.e. when $N = 0$ or $D = 0$).The potential at point P given a single triangle can now be defined as:

$$V(P, \Delta_{ABC}) = |I(A,B,G) + I(B,C,G) + I(C,A,G)| \tag{A.3}$$

For multiple triangles, the potential is simply the sum of individually computed potentials multiplied by the per-triangle charge density Q (as defined by the superposition principle).

$$V(P, T_{\text{obj}}, Q_{\text{obj}}) = \sum_{i=1}^{N_{\text{obj}}} V(P, \Delta^i_{\text{obj}}) Q^i_{\text{obj}} \tag{A.4}$$

A.2 Electric Field

The electric field for a single triangle is defined as the negative gradient of the electric potential. This can be analytically computed as the differentiated form of (A.3).

This is simply the negation of the sum of Jacobians of (A.2), i.e.:

$$E(P) = -(J(A,B,G) + J(B,C,G) + J(C,A,G)) \tag{A.5}$$

The analytical Jacobian of I(X,Y,G) is the sum of the gradients of each term in (A.2). We note that the gradients for term 2 and 3 are similar in form and can be expressed as a single function Ψ with parameters α and β:

$$\begin{aligned}\Psi(\alpha,\beta) =& \mathbf{n}_p \arctan(\frac{\alpha b \sigma}{v}) \\ &+ \frac{\mathbf{n}_p \cdot \beta}{\alpha^2 b^2 \sigma^2 + v^2} \left\{ \alpha\sigma(\alpha^2+\sigma^2) \left[\mathbf{n}_p |\beta| - \frac{\beta}{|\beta|}(\mathbf{n}_p \cdot \beta) \right] \right. \\ &\left. + b \left[\alpha(\vec{YX} \times \mathbf{n}) - \sigma \vec{YX} \right] \left[\alpha^2 (\mathbf{n}_p \cdot \beta) - \sigma^2 |\beta| \right] \right\}\end{aligned} \tag{A.6}$$

where $\mathbf{n}_p = \vec{GP}/|\vec{GP}|$ is the unit normal to Δ_{ABC} in the direction of point P, $v = \sigma^2|\beta| + \alpha^2(\mathbf{n}_p \cdot \beta))$ and $b = \mathbf{n}_p \cdot \beta - |\beta|$.

The parameters for term 2 are $\alpha := \vec{YX} \cdot \vec{YG}$, $\beta := \vec{YP}$, and for term 3 are $\alpha := \vec{YX} \cdot \vec{XG}$, $\beta := \vec{XP}$.

The equation of the gradient of the first term, Θ, is somewhat simpler:

$$\begin{aligned}\Theta =& \frac{\vec{YX} \times \mathbf{n}}{|\vec{YX}|} \log(\frac{N}{D}) \\ &+ \sigma \left[\left(\frac{\vec{YP}}{|\vec{YP}|} + \frac{\vec{YX}}{|\vec{YX}|} \right) / N - \left(\frac{\vec{XP}}{|\vec{XP}|} + \frac{\vec{YX}}{|\vec{YX}|} \right) / D \right]\end{aligned} \tag{A.7}$$

This gives the gradient of (A.2) as

$$I'(X,Y,G) = \Theta + \Psi(\vec{YX} \cdot \vec{YG}, \vec{YP}) - \Psi(\vec{YX} \cdot \vec{XG}, \vec{XP}) \tag{A.8}$$

A.3 Electric Flux

The flux is the measurement of the electric field through a surface. In this thesis the charged surface is defined by a closed mesh of triangles representing an object's surface, and the surface through which the flux is measured (henceforth called the *flux measurement surface*) is an arbitrary mesh of triangles that may be unconnected, partially connected, open or closed and consists of one or more triangles.

Due to the superposition principle overall flux Φ_{Flux} is computed via a computation that occurs between each individual triangle representing the object ($\Delta_{\text{obj}} \in T_{\text{obj}}$) and each individual triangle in the flux measurement surface ($\Delta_m \in T_{\text{m}}$), along with the surface charges computed for the object Q_{obj}.

$$\Phi_{\text{Flux}}(T_m, T_{\text{obj}}, Q_{\text{obj}}) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_{\text{obj}}} f_{\text{flux}}(\Delta_m^i, \Delta_{\text{obj}}^j) Q_{\text{obj}}^j \tag{A.9}$$

where f_{flux} is approximated by four point Gaussian quadrature of the object triangle as follows:

$$\begin{aligned} f_{\text{flux}}(\Delta_{ABC}, \Delta_{DEF}) = \frac{|\vec{DE} \times \vec{DF}|}{2} [&\phi(\Delta_{ABC}, x_1) + \phi(\Delta_{ABC}, x_2) \\ &+ \phi(\Delta_{ABC}, x_3) + \phi(\Delta_{ABC}, x_4)] \end{aligned} \tag{A.10}$$

where

$$\begin{aligned} x_1 = \frac{4D+E+F}{6}, x_2 = \frac{D+4E+F}{6}, \\ x_3 = \frac{D+E+4F}{6}, x_4 = \frac{D+E+F}{3} \end{aligned}$$

To compute the flux through a triangle Δ_{ABC} given point charge P an analytical solution has been defined by Van Oosterom and Strackee [1983]. Let $\mathbf{a} = \vec{AP}$, $\mathbf{b} = \vec{BP}$, $\mathbf{c} = \vec{CP}$. Finally, the flux $\phi(\Delta_{\text{ABC}}, P)$ is computed via

$$\tan\left(\frac{\phi(\Delta_{\text{ABC}}, P)}{2}\right) = \frac{(\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c}}{|\mathbf{a}||\mathbf{b}||\mathbf{c}| + (\mathbf{a} \cdot \mathbf{b})|\mathbf{c}| + (\mathbf{a} \cdot \mathbf{c})|\mathbf{b}| + (\mathbf{b} \cdot \mathbf{c})|\mathbf{a}|} \tag{A.11}$$

A.4 Jacobian for the Electric Flux

The Jacobian of the Electric Flux through a triangle Δ_{ABC} attached to a kinematic chain with configuration defined by $q \in \mathbb{R}^n$, caused by a charged triangle Δ_{DEF}, is given by the following:

$$\frac{\delta f_{\text{flux}}(\Delta_{ABC},\Delta_{DEF})}{\delta q} = \frac{|\vec{DE}\times\vec{DF}|}{2}\left[\frac{\phi(\Delta_{ABC},x_1)}{\delta q}+\frac{\phi(\Delta_{ABC},x_2)}{\delta q} + \frac{\phi(\Delta_{ABC},x_3)}{\delta q}+\frac{\phi(\Delta_{ABC},x_4)}{\delta q}\right] \tag{A.12}$$

$$\begin{aligned}
\frac{\phi(\Delta_{ABC},x)}{\delta q} =& 2(\frac{\delta\Upsilon}{\delta q}\Omega-\Upsilon\frac{\delta\Omega}{\delta q})(\Omega^2+\Upsilon^2)^{-1}, \text{where}\\
\frac{\delta\Upsilon}{\delta q} =& (\frac{\delta\mathbf{a}}{\delta q}\times\mathbf{b}+\mathbf{a}\times\frac{\mathbf{b}}{\delta q})\cdot\mathbf{c}+(\mathbf{a}\times\mathbf{b})\cdot\frac{\delta\mathbf{c}}{\delta q},\\
\frac{\delta\Omega}{\delta q} =& \frac{\delta|\mathbf{a}|}{\delta q}|\mathbf{b}||\mathbf{c}|+|\mathbf{a}|\frac{\delta|\mathbf{b}|}{\delta q}|\mathbf{c}|+|\mathbf{a}||\mathbf{b}|\frac{\delta|\mathbf{c}|}{\delta q}\\
&+(\frac{\delta\mathbf{a}}{\delta q}\mathbf{b}+\mathbf{a}\frac{\delta\mathbf{b}}{\delta q})|\mathbf{c}|+\mathbf{ab}\frac{\delta|\mathbf{c}|}{\delta q}\\
&+(\frac{\delta\mathbf{a}}{\delta q}\mathbf{c}+\mathbf{a}\frac{\delta\mathbf{c}}{\delta q})|\mathbf{b}|+\mathbf{a}\frac{\delta|\mathbf{b}|}{\delta q}\mathbf{c}\\
&+(\frac{\delta\mathbf{c}}{\delta q}\mathbf{b}+\mathbf{c}\frac{\delta\mathbf{b}}{\delta q})|\mathbf{a}|+\frac{\delta|\mathbf{a}|}{\delta q}\mathbf{bc}
\end{aligned} \tag{A.13}$$

where $\Omega(\Delta_{ABC},x)$ and $\Upsilon(\Delta_{ABC},x)$ are the numerator and denominator of the right-hand side of Equation A.11 respectively (the arguments have been dropped in Equation A.13 to aid clarity).

A.5 Time Complexity

The time complexity of computing the potential is $O(mn)$ where n is the number of sample points for which to compute and m is the number of triangles in the object mesh. Similarly, the flux computation time complexity is $O(lm)$ where l is the number of triangles in the flux measurement surface.

Appendix B

Description of the Interaction Database

During the creation of this thesis, an interaction database was created that contains scans of many objects and multiple takes using each object. The motions captured are enumerated below.

B.1 Captured Motions

- Ball - 1× repeated putting a ball into a box. 1× placing a ball into a box a single time. 1× throwing the ball. 3× picking it up from the ground and placing in different locations.
- Broom - 1× using broom to sweep floor. 1× marching whilst carrying broom.
- Book - 1× picking up the book.
- Bottle 1 (Lucozade) - 10× opening the bottle. 10× passing the bottle to someone else and receiving it back. 10× pouring out the contents of the bottle. 10× drinking from the bottle.
- Bottle 2 (large, 1 litre Lucozade) - 10× opening the bottle. 10× passing the bottle to someone else and receiving it back. 10× pouring out the contents of the bottle. 10× drinking from the bottle.
- Bottle 3 (Ribena) - 10× unscrew cap. 3× move bottle to new location. 5× drink from bottle. 2× long takes involving picking up, unscrewing the cap, drinking from the bottle, replacing the cap and returning the bottle to the ground.

- Box - 3× pickup and carry box.
- Bag and spray bottle - 1× repeatedly putting spray bottle into bag and removing from bag.
- Coffee jar - 1× repeated pickup, removal of lid (screw top), and replacement back onto coffee jar.
- Cup (medium) - 2× repeated pickup, drink with both one and two hands, using the handle and also not using the handle.
- Cup (small) - 1× as with 'Cup (medium)', using the handle.
- Cup (spots) - 10× drinking from the cup. 10× passing the cup to another person and receiving it back. 10× pouring the contents of the cup out. 10× stirring the cup.
- Cup (circles) - 10× drinking from the cup. 10× passing the cup to another person and receiving it back. 10× pouring the contents of the cup out. 10× stirring the cup.
- Cup (giraffe) - 10× drinking from the cup. 10× passing the cup to another person and receiving it back. 10× pouring the contents of the cup out. 10× stirring the cup.
- Clothes - 2× folding trousers, a hooded jumper and scarf. 1× dressing in clothes: putting on trousers, a hooded jumper and scarf. 1× tying a tie.
- Cereal box, bowl and spoon - 10× making breakfast (includes pouring cereal into bowl, eating using spoon, picking up and moving around each of the objects).
- Chair - 3× transitioning from standing to sitting on chair.
- Frog Toy - 7× moving the object to different locations, picking up the object.
- Hard Hat - 1× picking up hard hat and placing on ground. 2× picking up hard hat and wearing whilst walking.
- Knife - 10× chopping. 10× passing the knife to another person and receiving it back.
- Knife with 'orange' (ball) - 10× peeling an orange using the knife.
- Knife with two sections of an object - 10× cutting into an object, separating the object into two sections.
- Medicine container - 5× removing and replacing the lid. 4× moving the container to different locations on a table.
- Pan and spoon - 1× simulated cooking, stirring the pot, using the spoon to pick items out of the pot.

- Pill container - 7× removing the top of the container, moving the container to different locations.
- Pistol - 1× draw from holster, aim and fire in multiple directions using one hand, put back in to holster.
- Pen - 9× using the pen for writing and drawing.
- Rifle - 2× picking up from the floor, aim and fire in multiple directions with two hands, surrender, place back down on floor. 1× crouch and fire.
- Paper - 2× folding paper in different styles.
- Spray bottle - 3× long takes which involve picking up, moving, and using the spray bottle, as well as grasping the spray bottle in different locations, rotating, and placing sideways and flat on a table. 3× using a spray bottle, starting by picking it up in different styles and in different locations.
- Spoon and bowl - 10× eating from the bowl. 10× stirring the bowl. 9× longer takes of stirring using the spoon, eating from the bowl, picking up and moving the spoon and bowl around.
- Screwdriver - 9× using the screwdriver in different orientations
- Tennis Racquet - 1× picking up the racquet. 1× simulated tennis with the racquet.
- Table - 1× picking up table and carrying, then placing back on ground. 2× moving table to different locations.

There is a further small set of single takes without objects for comparison with the optical system that was captured during the creation of this thesis:

- Bending backwards
- Circling hips
- Crouching
- Crouching into a ball
- Crouching walk
- Exercising
- Kicking a leg
- Lifting a leg
- Marching
- Pointing at own head
- Performing a push up
- Pointing in different directions (arrow)
- Posing in different athletic styles

- Playing football
- Rolling the head
- Swinging the arms
- Turning around
- Testing range of arm motion
- T-Pose to walk left
- T-Pose to walk right
- T-Pose to walking backwards
- Walking forwards
- Walking forwards in a different style
- Walking to T-Pose

Bibliography

Al-Asqhar, R. A., Komura, T., and Choi, M. G. (2013). Relationship descriptors for interactive motion adaptation. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '13, pages 45–53, New York, NY, USA. ACM.

Albrecht, I., Haber, J., and Seidel, H. (2003). Construction and Animation of Anatomically Based Human Hand Models. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 98–109. Eurographics Association.

Aldebaran Robotics (2014). http://www.aldebaran.com/en. *website*.

Amenta, N. and Bern, M. (1999). Surface reconstruction by voronoi filtering. *Discrete & Computational Geometry*, 22(4):481–504.

Amenta, N., Choi, S., and Kolluri, R. K. (2001). The power crust. *Proceedings of the sixth ACM symposium on Solid modeling and applications - SMA '01*, pages 249–266.

Arbib, M., Iberall, T., and Lyons, D. (1985). Coordinated control programs for movements of the hand. *Hand Function and the Neocortex*, pages 111–129.

Aristidou, A. and Lasenby, J. (2010). Motion Capture with Constrained Inverse Kinematics for Real-Time Hand Tracking. *Proceedings of the 4th International Symposium on Communications, Control and Signal Procesing (ISCCSP)*, pages 1–5.

Aristidou, A. and Lasenby, J. (2011). Fabrik: a fast, iterative solver for the inverse kinematics problem. *Graphical Models*, 73(5):243–260.

Ascension (1998). Flock of birds real-time motion tracker. *Company Brochure, Ascension Technology Co., Burlington, VT*.

Autodesk, Inc. (2012). Autodesk FBX SDK. http://autodesk.com/fbx. Date accessed: June 2012.

Aydin, Y. (1999). Database guided computer animation of human grasping using forward and inverse kinematics. *Computers & Graphics*, 23(1):145–154.

Barile, M. and Weisstein, E. W. (2014). Betti number. *[Online]: From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/BettiNumber.html, accessed 20 July 2014*.

Bekey, G., Tomovic, R., and Karplus, W. (1993). Knowledge-based control of grasping in robot hands using heuristics from human motor skills. *IEEE Transactions on Robotics and Automation*, 9(6):709–722.

Belter, J. and Dollar, A. (2011). Performance characteristics of anthropomorphic prosthetic hands. In *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*, pages 1–7. IEEE.

Berenstein, C. A. and Gay, R. (1997). *Complex Variables: An Introduction*. Springer.

Berger, M., Levine, J. A., Nonato, L. G., Taubin, G., and Silva, C. T. (2013). A benchmark for surface reconstruction. *ACM Trans. Graph.*, 32(2):20:1–20:17.

Bicchi, A. and Kumar, V. (2000). Robotic grasping and contact: A review. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 348–353, San Francisco, CA.

Bierbaum, A., Rambow, M., Asfour, T., and Dillmann, R. (2009). Grasp affordances from multi-fingered tactile exploration using dynamic potential fields. In *Proc. of Humanoids*, pages 168–174.

Billard, A., Calinon, S., Dillmann, R., and Schaal, S. (2008). Robot programming by demonstration. In *Springer handbook of robotics*, pages 1371–1394. Springer.

Bittar, E., Tsingos, N., and Gascuel, M.-P. (1995). Automatic reconstruction of unstructured 3d data: Combining a medial axis and implicit surfaces. *Computer Graphics Forum*, 14(3):457–468.

Blinn, J. F. (1982). A generalization of algebraic surface drawing. *ACM Trans. Graph.*, 1(3):235–256.

Blood, E. (1989). Device for quantitatively measuring the relative position and orientation of two bodies in the presence of metals utilizing direct current magnetic fields. *U.S. Patent 4,849,692, July 18*.

Boulic, R., Rezzonico, S., and Thalmann, D. (1996). Multi-finger manipulation of virtual objects. In *In Proc. of the ACM Symposium on Virtual Reality Software and Technology (VRST '96*, pages 67–74.

Bronstein, A. M., Bronstein, M. M., and Kimmel, R. (2008). *Numerical geometry of non-rigid shapes*. Springer.

Brown, E., Rodenberg, N., Amend, J., Mozeika, A., Steltz, E., Zakin, M. R., Lipson, H., and Jaeger, H. M. (2010). Universal robotic gripper based on the jamming of granular material. *Proceedings of the National Academy of Sciences*, 107(44):18809–18814.

Burdea, G. (1993). Virtual reality systems and applications [short course]. *Electro '93 International Conference*.

Burns, E., Razzaque, S., Panter, A., Whitton, M., McCallus, M., and Brooks, F. (2006). The hand is more easily fooled than the eye: Users are more sensitive to visual interpenetration than to visual-proprioceptive discrepancy. *Presence*, 15(1):1–15.

Burrus, N. (2012). Rgbdemo. `http://labs.manctl.com/rgbdemo/`. Date accessed: June 2012.

Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C., and Evans, T. R. (2001). Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76. ACM.

Carr, J. C., Beatson, R. K., McCallum, B. C., Fright, W. R., McLennan, T. J., and Mitchell, T. J. (2003). Smooth surface reconstruction from noisy range data. In *Proceedings of the 1st International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*, GRAPHITE '03, pages 119–ff, New York, NY, USA. ACM.

Catmull, E. (1972). A system for computer generated movies. In *Proceedings of the ACM annual conference-Volume 1*, pages 422–431. ACM.

Ciocarlie, M., Dang, H., Lukos, J., Santello, M., and Allen, P. (2009). Functional analysis of finger contact locations during grasping. *World Haptics 2009 - Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 401–405.

Ciocarlie, M., Goldfeder, C., and Allen, P. (2007). Dimensionality reduction for hand-independent dexterous robotic grasping. In *Proc. of IROS*, pages 3270–3275.

Cui, J. and Sun, Z. (2004). Visual hand motion capture for guiding a dexterous hand. *Sixth IEEE International Conference on Automatic Face and Gesture Recognition (Proceedings)*, pages 729–734.

Cutkosky, M. R. and Howe, R. D. (1990). Human grasp choice and robotic grasp analysis. *Dextrous Robot Hands*, pages 5–31.

Davis, J., Marschner, S. R., Garr, M., and Levoy, M. (2002). Filling holes in complex surfaces using volumetric diffusion. In *3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on*, pages 428–441. IEEE.

Delfinado, C. J. A. and Edelsbrunner, H. (1993). An incremental algorithm for betti numbers of simplicial complexes. In *Proceedings of the Ninth Annual Symposium on Computational Geometry*, SCG '93, pages 232–239, New York, NY, USA. ACM.

Dey, T. K. (2007). *Curve and surface reconstruction: algorithms with mathematical analysis*. Cambridge University Press.

Dey, T. K., Fan, F., and Wang, Y. (2013). An efficient computation of handle and tunnel loops via reeb graphs. *ACM Trans. Graph.*, 32(4):32:1–32:10.

Diankov, R., Srinivasa, S. S., Ferguson, D., and Kuffner, J. (2008). Manipulation planning with caging grasps. In *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, pages 285–292. IEEE.

Do, M., Azad, P., Asfour, T., and Dillmann, R. (2008). Imitation of human motion on a humanoid robot using non-linear optimization. In *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, pages 545–552. IEEE.

El-Khoury, S., Li, M., and Billard, A. (2012). Bridging the gap: One shot grasp synthesis approach. In *IEEE/RSJ IROS*, pages 2027–2034.

ElKoura, G. and Singh, K. (2003). Handrix: animating the human hand. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '03, pages 110–119, Aire-la-Ville, Switzerland. Eurographics Association.

Elliott, J. M. and Connolly, K. J. (1984). A classification of manipulative hand movements. pages 283–96.

Erol, A., Bebis, G., Nicolescu, M., Boyle, R. D., and Twombly, X. (2007). Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108(1-2):52–73.

Ferrari, C. and Canny, J. (1992). Planning optimal grasps. *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pages 2290–2295.

Fifth Dimension Technologies (2014). 5DT Dataglove http://www.5dt.com.

Formaggio, J. A., Lazic, P., Corona, T., Stefancic, H., Abraham, H., and Gluck, F. (2011). Solving for micro-and macro-scale electrostatic configurations using the robin hood algorithm. *arXiv preprint arXiv:1111.5035*.

Gaiser, I., Schulz, S., Kargov, A., Klosek, H., Bierbaum, A., Pylatiuk, C., Oberle, R., Werner, T., Asfour, T., Bretthauer, G., and Dillmann, R. (2008). A new anthropomorphic robotic hand. In *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on*, pages 418–422.

Giansanti, D., Maccioni, G., and Macellari, V. (2005). The development and test of a device for the reconstruction of 3-D position and orientation by means of a kinematic sensor assembly with rate gyroscopes and accelerometers. *IEEE Trans Biomed Eng*, 52(7):1271–1277.

Gioioso, G., Salvietti, G., Malvezzi, M., and Prattichizzo, D. (2013). Mapping synergies from human to robotic hands with dissimilar kinematics: An approach in the object domain. *Robotics, IEEE Transactions on*, 29(4):825–837.

Giorgi, D., Biasotti, S., and Paraboschi, L. (2007). Shrec: shape retrieval contest: Watertight models track. *[Online]: `http://watertight.ge.imati.cnr.it`, accessed June 2014*.

Gleicher, M. and Litwinowicz, P. (1998). Constraint-based motion adaptation. *The Journal of Visualization and Computer Animation*, 9(2):65–94.

Goto, E., Shi, Y., and Yoshida, N. (1992). Extrapolated Surface Charge Method for Capacity Calculation of Polygons and Polyhedra. In *Journal of Computational Physics*, volume 100, pages 105–115.

Grollman, D. H. and Jenkins, O. C. (2007). Learning robot soccer from demonstration: Ball grasping. In *Robotics: Science and Systems - Robot Manipulation: Sensing and Adapting to the Real World*.

Haex, M. and Gambardella, L. M. (1992). Stable Grasps by Path Planning using Artificial Fields. In *Proceedings of the 1992 lEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1664–1669.

Hamer, H., Gall, J., and Urtasun, R. (2011). Data-driven animation of hand-object interactions. *Automatic Face and Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 360–367.

Hamer, H., Gall, J., Urtasun, R., and Van Gool, L. (2010a). Animating Hand-Object Interaction. Technical report.

Hamer, H., Gall, J., Weise, T., and Van Gool, L. (2010b). An object-dependent hand pose prior from sparse training data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 671–678. IEEE.

Hamer, H., Schindler, K., Koller-Meier, E., and Van Gool, L. (2009). Tracking a hand manipulating an object. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1475–1482.

Hashimoto, H., Sasaki, A., Yokota, S., Ohyama, Y., and Ishii, C. (2011). A study on degree of freedom in hand modeling. In *SICE Annual Conference (SICE), 2011 Proceedings of*, pages 2492–2493.

Hertkorn, K., Weber, B., Kremer, P., Roa, M. A., and Borst, C. (2013). Assistance for telepresence using online grasp planning. *IEEE/RAS International Conference on Humanoid Robotics*.

Hilaga, M., Shinagawa, Y., Kohmura, T., and Kunii, T. L. (2001). Topology matching for fully automatic similarity estimation of 3D shapes. In *Proceedings of the 28th*

annual conference on Computer graphics and interactive techniques - SIGGRAPH '01, pages 203–212, New York, New York, USA. ACM Press.

Hillenbrand, U. and Roa, M. (2012). Transferring functional grasps through contact warping and local replanning. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2963–2970.

Ho, E. S. L. and Komura, T. (2009). Character Motion Synthesis by Topology Coordinates. In *Computer Graphics Forum (Proc. Eurographics 2009)*, volume 28, Munich, Germany.

Ho, E. S. L., Komura, T., and Tai, C.-L. (2010). Spatial relationship preserving character motion adaptation. *ACM Transactions on Graphics*, 29(4):1–8.

Hu, H., Li, J., Xie, Z., Wang, B., Liu, H., and Hirzinger, G. (2005). A robot arm/hand teleoperation system with telepresence and shared control. In *Proc. of AIM*, pages 1312 –1317.

Huang, H., Zhao, L., Yin, K., Qi, Y., and Yu, Y. (2011). Controllable hand deformation from sparse examples with rich details. *Proceedings of SCA 2011*, 1:73–82.

Huang, Z., Boulic, R., Thalmann, N., and Thalmann, D. (1995). A multi-sensor approach for grasping and 3D interaction. *Computer Graphics International95*.

Huebner, K. and Kragic, D. (2008). Selection of robot pre-grasps using box-based shape approximation. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1765–1770.

Iberall, T., Bingham, G., and Arbib, M. (1986). Opposition space as a structuring concept for the analysis of skilled hand movements. *Experimental Brain Research Series 15 - Generation and Modulation of Action Patterns*, pages 158–173.

Igarashi, Y. and Suzuki, H. (2011). Cover geometry design using multiple convex hulls. *Comput. Aided Des.*, 43(9):1154–1162.

Ivan, V., Zarubin, D., Toussaint, M., Komura, T., and Vijayakumar, S. (2013). Topology-based representations for motion planning and generalisation in dynamic environments with interactions. *The International Journal of Robotics Research*.

Johansson, R. (1996). Sensory and memory information in the control of dexterous manipulation. In Lacquaniti, F. and Viviani, P., editors, *Neural Bases of Motor Behaviour*, volume 85 of *NATO ASI Series*, pages 205–260. Springer Netherlands.

Jörg, S., Hodgins, J., and O'Sullivan, C. (2010). The perception of finger motions. In *Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization*, volume 1, pages 129–133. ACM.

Kaick, O. V., Zhang, H., Hamarneh, G., and Cohen-or, D. (2011). A Survey on Shape Correspondence. *Computer Graphics Forum*, 30(6):1681–1707.

Kang, S. and Ikeuchi, K. (1997). Toward automatic robot instruction from perception – Mapping human grasps to manipulator grasps. *IEEE Trans. on Robotics and Automation*, 13(1).

Kang, S. B. and Ikeuchi, K. (1992). Grasp recognition using the contact web. *Proc. IEEE/RSJ Int. Conf. on Int. Robots and Sys., Raleigh, NC*.

Kang, S. B. and Ikeuchi, K. (1994). Robot Task Programming by Human Demonstration: Mapping Human Grasps to Manipulator Grasps. In *Proc. of IROS*, volume 1, pages 97–104.

Kavraki, L., Svestka, P., Latombe, J.-C., and Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Robotics and Automation, IEEE Transactions on*, 12(4):566–580.

Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, pages 61–70, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Kessler, G. D., Hodges, L. F., and Walker, N. (1995). Evaluation of the cyberglove as a whole-hand input device. *ACM Trans. Comput.-Hum. Interact.*, 2(4):263–283.

Khatib, O. (1986). Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *International Journal of Robotics Research*, 5(1):90–98.

Khoshelham, K. and Elberink, S. O. (2012). Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454.

Kim, S., Shukla, A., and Billard, A. (2014). Catching objects in flight. In *IEEE Transactions on Robotics*, volume 30, pages 1049–1065.

Kim, V. G., Lipman, Y., and Funkhouser, T. (2011). Blended intrinsic maps. In *ACM SIGGRAPH 2011 papers*, SIGGRAPH '11, pages 79:1–79:12, New York, NY, USA. ACM.

Klein, C. and Huang, C.-H. (1983). Review of pseudoinverse control for use with kinematically redundant manipulators. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-13(2):245–250.

Koga, Y., Kondo, K., Kuffner, J., and Latombe, J. (1994). Planning Motions with Intentions. SIGGRAPH '94, pages 395–408, New York, NY, USA. ACM.

Krieg, J. (1993). Motion tracking: Polhemus technology. *Virtual Reality Systems*, 1(1):32–36.

Kry, P. G. and Pai, D. K. (2006). Interaction capture and synthesis. *ACM Transactions on Graphics*, 25(3):872–880.

Kuperberg, W. (1990). Problems on polytopes and convex sets. In *Workshop on Polytopes*.

Latash, M. L., Scholz, J. P., and Schöner, G. (2007). Toward a new theory of motor synergies. *Motor control*, 11(3):276–308.

LaValle, S. M. (1998). Rapidly-Exploring Random Trees: A New Tool for Path Planning.

Lederman, S. J. and Klatzky, R. L. (1987). Hand movements: A window into haptic object recognition. *Cognitive psychology*, 19(3):342–368.

León, B., Ulbrich, S., Diankov, R., Puche, G., Przybylski, M., Morales, A., Asfour, T., Moisio, S., Bohg, J., Kuffner, J., and Dillmann, R. (2010). Opengrasp: A toolkit for robot grasping simulation. In *SIMPAR*, volume 6472 of *Lecture Notes in Computer Science*, pages 109–120. Springer Berlin Heidelberg.

Li, Y., Fu, J., and Pollard, N. (2007). Data-driven grasp synthesis using shape matching and task-based pruning. In *IEEE transactions on visualization and computer graphics*, volume 13, pages 732–747.

Lipman, Y. and Funkhouser, T. (2009). Möbius voting for surface correspondence. *ACM Transactions on Graphics*, 28(3):1.

Liu, C. K. (2008). Synthesis of Interactive Hand Manipulation. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*.

Liu, C. K. (2009). Dextrous Manipulation from a Grasping Pose. *ACM Transactions on Graphics (SIGGRAPH)*, 28(3).

Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pages 163–169, New York, NY, USA. ACM.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157 vol.2.

MacKenzie, C. L. and Iberall, T. (1994). *The Grasping Hand*. North-Holland, Elsevier Science B.V.

Masehian, E. and Katebi, Y. (2007). Robot Motion Planning in Dynamic Environments with Moving Obstacles and Target. *Engineering and Technology*, pages 107–112.

Merrell, P., Schkufza, E., Li, Z., Agrawala, M., and Koltun, V. (2011). Interactive furniture layout using interior design guidelines. *ACM Trans. Graph.*, 30(4):87:1–87:10.

Michelman, P. and Allen, P. (1994). Forming complex dextrous manipulations from task primitives. *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 3383–3388.

Miller, A. and Allen, P. (1999). Examples of 3D grasp quality computations. In *Proc. of ICRA*, volume 2, pages 1240–1246.

Miller, A. and Allen, P. (2004). Graspit! a versatile simulator for robotic grasping. *IEEE Robotics Aut. Mag.*, 11(4):110–122.

Mitobe, K., Kaiga, T., Yukawa, T., and Miura, T. (2006). Development of a motion capture system for a hand using a magnetic three dimensional position sensor. *ACM SIGGRAPH Research posters*, page 1.

Molet, T., Boulic, R., Rezzonico, S., and Thalmann, D. (1999a). An architecture for immersive evaluation of complex human tasks. *IEEE Transaction in Robotics and Automation, Special Section on Virtual Reality*, 15(3):475–485.

Molet, T., Boulic, R., and Thalmann, D. (1999b). Human motion capture driven by orientation measurements. *Presence: Teleoperators and Virtual Environments*, 8(2):187 – 203.

Mordatch, I. and Todorov, E. (2012). Contact-Invariant Optimization for Hand Manipulation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '12, pages 137–144, Aire-la-Ville, Switzerland. Eurographics Association.

Munoz-Salinas, R. (2012). Aruco: a minimal library for augmented reality applications based on opencv. `http://www.uco.es/investiga/grupos/ava/node/26`. Date accessed: June 2012.

Murray, R. M., Li, Z., and Sastry, S. S. (2006). *A Mathematical Introduction to Robotic Manipulation*. CRC Press.

Nakaoka, S. and Komura, T. (2012). Interaction mesh based motion adaptation for biped humanoid robots. In *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, pages 625–631.

Napier, J. (1956). The Prehensible Movements of the Human Hand. *The Journal of Bone and Joint Surgery*, 38B(4):902–913.

Naturalpoint Inc. (2014). OptiTrack FLEX:V100 Overview. `http://www.naturalpoint.com/optitrack/products/flex-v100/`. Date accessed: January 2014.

O'Donovan, K. J., Kamnik, R., O'Keeffe, D. T., and Lyons, G. M. (2007). An inertial and magnetic sensor based technique for joint angle measurement. *Journal of Biomechanics*, 40(12):2604–2611.

Ogawara, K., Takamatsu, J., Kimura, H., and Ikeuchi, K. (2003). Extraction of essential interactions through multiple observations of human demonstrations. *Industrial Electronics, IEEE Transactions on*, 50(4):667–675.

Oikonomidis, I., Kyriazis, N., and Argyros, A. (2011a). Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2088–2095.

Oikonomidis, I., Kyriazis, N., and Argyros, A. A. (2011b). Markerless and Efficient 26-DOF Hand Pose Recovery. In *Proceedings of the 10th Asian Conference on Computer Vision - Volume Part III*, ACCV'10, pages 744–757, Berlin, Heidelberg. Springer-Verlag.

Ovsjanikov, M., Mérigot, Q., Mémoli, F., and Guibas, L. (2010). One point isometric matching with the heat kernel. In *Computer Graphics Forum*, volume 29, pages 1555–1564. Wiley Online Library.

Pao, L. and Speeter, T. H. (1989). Transformation of human hand positions for robotic hand control. In *Proc. of RA*, pages 1758 –1763 vol.3.

Peer, A., Einenkel, S., and Buss, M. (2008). Multi-fingered telemanipulation - mapping of a human hand to a three finger gripper. In *Proc. of RO-MAN*, pages 465 –470.

Pokorny, F. T. and Kragic, D. (2013). Classical grasp quality evaluation: New theory and algorithms. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan.

Pokorny, F. T., Stork, J. A., and Kragic, D. (2013). Grasping objects with holes: A topological approach. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany.

Pollard, N. S. and Zordan, V. B. (2005). Physically based grasping control from example. *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation - SCA '05*, (July):311–318.

Quek, F. K. H. (1996). Unencumbered gestural interaction. *IEEE MultiMedia*, 3(4):36–47.

Rahman, M. M., Mitobe, K., Suzuki, M., Takano, C., and Yoshimura, N. (2011). Analysis of dexterous finger movement for piano education using motion capture system. *Internation Journal of Science and Technology Education Research*.

Rawlik, K., Toussaint, M., and Vijayakumar, S. (2013). On Stochastic Optimal Control and Reinforcement Learning by Approximate Inference. In *Proceedings of the*

Twenty-Third International Joint Conference on Artificial Intelligence, IJCAI '13, pages 3052–3056. AAAI Press.

Reichel, M. and The Shadow Robot Company (2004). Transformation of Shadow Dextrous Hand and Shadow Finger Test Unit from Prototype to Product for Intelligent Manipulation and Grasping. Date accessed: July 2011.

Rezzonico, S., Huang, Z., Boulic, R., Magnenat Thalmann, N., and Thalmann, D. (1995). Consistent grasping interactions with virtual actors based on the multi-sensor hand model. In *Virtual environments '95 : selected papers of the Eurographics workshops in Barcelona, Spain, 1993 and Monte Carlo, Monaco, 1995*, volume 30.

Roa, M. and Suarez, R. (2009). Computation of Independent Contact Regions for Grasping 3-D Objects. *IEEE Transactions on Robotics*, 25(6):839–850.

Roa, M. A., Argus, M. J., Leidner, D., Borst, C., and Hirzinger, G. (2012). Power grasp planning for anthropomorphic robot hands. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 563–569. IEEE.

Roa, M. A., Hertkorn, K., Borst, C., and Hirzinger, G. (2011). Reachable independent contact regions for precision grasps. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5337–5343. IEEE.

Rodriguez, A., Mason, M. T., and Ferry, S. (2011). From Caging to Grasping. In *Robotics: Science and Systems (RSS)*.

Roetenberg, D., Luinge, H., and Slycke, P. (2013). Xsens MVN: Full 6DOF Human Motion Tracking Using Miniature Inertial Sensors.

Rusu, R. B., Blodow, N., and Beetz, M. (2009). Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 3212 –3217.

Saha, M. and Isto, P. (2007). Manipulation Planning for Deformable Linear Objects. *IEEE Transactions on Robotics*, 23(6):1141–1150.

Sandilands, P., Choi, M. G., and Komura, T. (2012). Capturing Close Interactions with Objects Using a Magnetic Motion Capture System and a RGBD Sensor. In *Motion in Games*, volume 7660, pages 220–231. Springer Berlin Heidelberg.

Sandilands, P., Choi, M. G., and Komura, T. (2013a). Interaction Capture using Magnetic Sensors. *Computer Animation and Virtual Worlds*, 24(6):527–538.

Sandilands, P., Ivan, V., Komura, T., and Vijayakumar, S. (2013b). Dexterous Reaching, Grasp Transfer and Planning Using Electrostatic Representations. *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*.

Sanso, R. M. and Thalmann, D. (1994). A Hand Control and Automatic Grasping System for Synthetic Actors. *Computer Graphics Forum*, 13(3):167–177.

Santello, M., Flanders, M., and Soechting, J. F. (1998). Postural hand synergies for tool use. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 18(23):10105–15.

Sauser, E. L., Argall, B. D., Metta, G., and Billard, A. G. (2012). Iterative learning of grasp adaptation through human corrections. *Robotics and Autonomous Systems*, 60(1):55–71.

SCHUNK GmbH and Co (2012). http://mobile.schunk-microsite.com/en/produkte/produkte/sdh-servo-electric-3-finger-gripping-hand.html. *website*.

Slocum, D. B. and Pratt, D. R. (1946). Disability evaluation for the hand. *Journal of Bone and Joint Surgery*.

Song, P. and Kumar, V. (2002). A potential field based approach to multi-robot manipulation. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 2, pages 1217–1222.

Steffen, J., Haschke, R., and Ritter, H. (2008). Using Manifolds for Dextrous Hand Control. In *RSS 2008 Workshop on Robot Manipulation: Intelligence in Human Environments*.

Stork, J. A., Pokorny, F. T., and Kragic, D. (2013). A topology-based object representation for clasping, latching and hooking. In *IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)*, Atlanta, USA.

Suárez, R., Roa, M., and Cornella, J. (2006). Grasp quality measures. Technical report, Universitat Polytècnica de Catalunya.

Sueda, S., Kaufman, A., and Pai, D. K. (2008). Musculotendon simulation for hand animation. *ACM Transactions on Graphics*, 27(3):8.

Takahashi, S., Ikeda, T., Shinagawa, Y., Kunii, T. L., and Ueda, M. (1995). Algorithms for extracting correct critical points and constructing topological graphs from discrete geographical elevation data. In *Computer Graphics Forum*, volume 14, pages 181–192. Wiley Online Library.

Tatematsu, A., Hamada, S., and Takuma, T. (2002). Analytical expressions of potential and electric field generated by a triangular surface charge with a high-order charge density distribution. *Electrical Engineering in Japan*, 139(3):9–17.

Taylor, C. L. and Schwarz, R. J. (1955). The anatomy and mechanics of the human hand. *Artificial Limbs: A Review of Current Developments*, 2(2).

Tegin, J., Iliev, B., Skoglund, A., Kragic, D., and Wikander, J. (2009). Real Life Grasping using an Under-actuated Robot Hand Simulation and Experiments. *Design*, pages 1–8.

Tipler, P. A. and Mosca, G. (2007). *Physics for scientists and engineers*. Macmillan.

Toussaint, M. (2009). Robot Trajectory Optimization using Approximate Inference. In *Proc. of ICML*, pages 1049–1056. ACM.

Ude, A., Atkeson, C. G., and Riley, M. (2004). Programming full-body movements for humanoid robots by observation. *Robotics and Autonomous Systems*, 47(2):93–108.

Vahrenkamp, N., Kröhnert, M., Ulbrich, S., Asfour, T., Metta, G., Dillmann, R., and Sandini, G. (2012). Simox: A robotics toolbox for simulation, motion and grasp planning. In *IEEE Int. Conf. on Intelligent Autonomous Systems (IAS)*.

van Nierop, O. a., van der Helm, A., Overbeeke, K. J., and Djajadiningrat, T. J. (2007). A natural human hand model. *The Visual Computer*, 24(1):31–44.

Van Oosterom, A. and Strackee, J. (1983). The Solid Angle of a Plane Triangle. *IEEE Transactions on Biomedical Engineering*, BME-30(2):125–126.

Vlasic, D., Adelsberger, R., Vannucci, G., Barnwell, J., Gross, M., Matusik, W., and Popović, J. (2007). Practical motion capture in everyday surroundings. *ACM Trans. Graph.*, 26(3).

Wakamatsu, H., Tsumaya, A., Arai, E., and Hirai, S. (2005). Manipulation Planning for Knotting/Unknotting and Tightly Tying of Deformable Linear Objects. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2516–2521.

Walairacht, S., Yamada, K., Hasegawa, S., Koike, Y., and Sato, M. (2004). Two-handed multiple-finger virtual object manipulation environment with haptic cues. *Electronics and Communications in Japan (Part II: Electronics)*, 87(9):65–73.

Wang, H. and Komura, T. (2011). Energy-Based pose unfolding and interpolation for 3D articulated characters. In *Motion in Games*, volume 7060, pages 110–119. Springer Berlin Heidelberg, Berlin, Heidelberg.

Wang, H. and Komura, T. (2012). Manipulation of Flexible Objects by Geodesic Control. *Comp. Graph. Forum*, 31(2pt2):499–508.

Wang, H., Sidorov, K. A., Sandilands, P., and Komura, T. (2013a). Harmonic Parameterization by Electrostatics. *ACM Trans. Graph.*, 32(5):155:1–155:12.

Wang, R. and Popovic, J. (2009). Real-time hand-tracking with a color glove. In *ACM SIGGRAPH 2009 Papers*, SIGGRAPH '09, pages 63:1–63:8, New York, NY, USA. ACM.

Wang, Y., Min, J., Zhang, J., Liu, Y., Xu, F., Dai, Q., and Chai, J. (2013b). Video-based Hand Manipulation Capture Through Composite Motion Control. *ACM Trans. Graph.*

Wei, G., Dai, J., Wang, S., and Luo, H. (2011a). Kinematic Analysis and Prototype of a Metamorphic Anthropomorphic Hand with a Reconfigurable Palm. *International Journal Of Humanoid Robotics*, 8(3):459 – 479.

Wei, X., Min, J., and Chai, J. (2011b). Physically valid statistical models for human motion generation. *ACM Trans. Graph.*, 30(3):19:1–19:10.

Weiss, E. J. and Flanders, M. (2004). Muscular and postural synergies of the human hand. *Journal of neurophysiology*, 92(1):523–35.

Xie, H., Xu, W., and Wang, B. (2013). Reshuffle-based interior scene synthesis. In *Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, VRCAI '13, pages 191–198, New York, NY, USA. ACM.

Yamane, K., Kuffner, J. J., and Hodgins, J. K. (2004). Synthesizing animations of human manipulation tasks. *ACM Transactions on Graphics*, 23(3):532.

Yasumuro, Y. and Chihara, K. (1997). 3D modeling of human hand with motion constraints. *Proceedings. International Conference on Recent Advances in 3-D Digital Imaging and Modeling (Cat. No.97TB100134)*, pages 275–282.

Ye, Y. and Liu, C. K. (2012). Synthesis of detailed hand manipulations using contact sampling. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2012)*, 31(4):41:1–41:10.

Zarubin, Dmitry, A., Pokorny, F. T., Toussaint, M., and Kragic, D. (2013). Caging complex objects with geodesic balls. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan.

Zarubin, D., Ivan, V., Toussaint, M., Komura, T., and Vijayakumar, S. (2012). Hierarchical Motion Planning in Topological Representations. In *Robotics: Science and Systems*.

Zhao, W., Chai, J., and Xu, Y.-Q. (2012). Combining marker-based mocap and rgb-d camera for acquiring high-fidelity hand motion data. *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation*.

Zhao, W., Zhang, J., Min, J., and Chai, J. (2013). Robust realtime physics-based motion control for human grasping. *ACM Trans. Graph.*, 32(6):207:1–207:12.

Zordan, V. B. and Hodgins, J. K. (2002). Motion capture-driven simulations that hit and react. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '02, pages 89–96, New York, NY, USA. ACM.